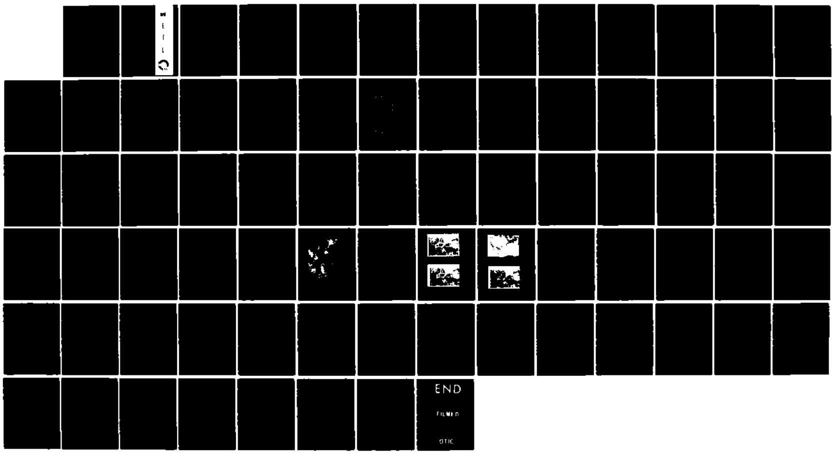
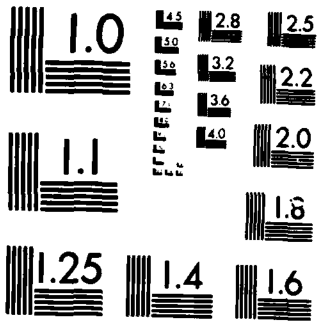
AD-A150 176 HEXAGONAL DATA BASE STUDY(U) INTERACTIVE SYSTEMS CORP LITTLETON CO L GIBSON ET AL. OCT 84 ETL-0360 DAAK70-82-C-0133 1/1

UNCLASSIFIED F/G 9/2 NL




1.0
4.5
5.0
5.6
6.3
2.8
2.5
3.2
2.2
3.6
4.0
2.0
1.1
1.8
1.25
1.4
1.6

ETL-360

6

Hexagonal data base study, Phase II

Laurie Gibson
William Gomer
Dean Lucas

Interactive Systems Corporation
5500 South Sycamore Street
Littleton, Colorado 80120

October 1984

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION IS UNLIMITED.

Prepared for

U.S. ARMY CORPS OF ENGINEERS
ENGINEER TOPOGRAPHIC LABORATORIES
FORT BELVOIR, VIRGINIA 22060-5546

E T L

U.S. ARMY ENGINEER TOPOGRAPHIC LABORATORIES
ETL

HEXAGONAL DATA BASE STUDY

PHASE II

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE		READ INSTRUCTIONS BEFORE COMPLETING FORM
1. REPORT NUMBER ETL-0360	2. GOVT ACCESSION NO. AD-A150176	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) Hexagonal Data Base Study, Phase II		5. TYPE OF REPORT & PERIOD COVERED Final Report October 1983 - October 1984
		6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s) Laurie Gibson, William Gomer, Dean Lucas		8. CONTRACT OR GRANT NUMBER(s) DAAK70-82-C-0133
9. PERFORMING ORGANIZATION NAME AND ADDRESS Interactive Systems Corporation 5500 South Sycamore Street Littleton, Colorado 80120		10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS U.S. Army Engineer Topographic Laboratories Fort Belvoir, Virginia 22060-5546		12. REPORT DATE October, 1984
		13. NUMBER OF PAGES
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)		15. SECURITY CLASS. (of this report) Unclassified
		15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

The first phase of this study was published September 1983 as ETL-0338 (AD-A135 532).

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Hexagonal data structures, automated mapping, polygon processing

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This is a final report for a study in the application of hexagonal data structures to handling geographic information. This report presents the results of an experimental software implementation utilizing hexagonal data structures as applied to test data base.

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

PREFACE

This report was produced under contract DAAK 70-82-C-0133 for the U.S. Army Engineer Topographic Laboratories (ETL), Fort Belvoir, Virginia 22060-5546. The Contracting Officer's Representative was Mr. Carl S. Huzzen. Principal Investigator for this study was Mr. William Gomer of Interactive Systems Corporation.

TABLE OF CONTENTS

1.0 **INTRODUCTION** 1

2.0 **THEORY AND IMPLMENTATION** 2
2.1 Overview 2
2.2 Nodes and Relations 2
2.3 The Addressing Scheme 4
2.4 The Data Tree 8
2.5 Searches 10
2.6 Attribute Data Management 10
2.7 AGIS - Linking RIM and GRAM 14
2.8 Point Data 15
2.9 Line Date 15
2.10 Region Data 15
2.11 Topology Verification 20
2.12 Operations on Map Layers 26
2.12.1 Operations on Regions and Regions 33
2.12.2 Operations on Regions and Lines 34
2.12.3 Operations on Regions and Points 34

3.0 **IMAGE HANDLING** 37
3.1 The Need 37
3.2 Imagery and Picture Files 37
3.3 Registration 39
3.4 Display 39

4.0 **WORKING WITH THE SYSTEM** 42
4.1 Overview 42
4.2 AGIS 42
4.3 Input 42
4.3.1 Map File Creation 42
4.3.2 Manual Digitization 42
4.3.3 Verification of Region Topology 43
4.3.4 Attribute Assignment 43
4.3.5 Database Statistics 54
4.4 Query Functions 56

5.0 **RESULTS AND CONCLUSIONS** 66
5.1 Overview 66
5.2 Integrate Point, Line, and Region Data 66
5.3 Automatic Association of Islands 66
5.4 Imagery 67

REFERENCES 68

List of Figures

2.1 The functional relationship between the AGIS subsystems 3
2.2 Node Connectivity 5
2.3 Node sets representing map data 6
2.4 The aggregate structure............... 7
2.5 The aggreagate 377 is higher than 327............... 9
2.6 The tree generated by a set of GBT addresses 9
2.7 A GRAM Tree 11
2.8 Area Search 12
2.9a General tree node data record 13
2.9b Tree node with Boolean field used for node data type & degree 13
2.10 The node network is stored in a GRAM file 16
2.11 Point Signpost 17
2.12 Line Signposts 18
2.13 Regional Signposts 19
2.14 The right-hand rule orients the boundaries and permits the correct signpost to be identified............... 21
2.15 Regions with Disconnected Boundaries 22
2.16 Isthmuses connect disconnected boundaries 23
2.17 A region layer that is not topologically correct............... 24
2.18 Line Crossing Algorithm checks to see if CD_1 or CD_2 cross AB 25
2.19 A Network with Four Components 27
2.20 Bounding rectangles overlap, nest, or are disjoint 28
2.21 Building an Isthmus from Component A............... 29
2.22 A Node Network with Isthmuses and a Frame 30
2.23 Query Bitmap Generation 35
2.24 Lines clipped to the interiors of regions 36
3.1 A portion of an Aerial Photograph of the Healdsburg Region 38
3.2 A Picture File Displayed by AGIS............... 40
3.3 The Same Picture with Vectors Overlaid............... 40
3.4 The Vector File from Figure 3.3 without the Picture 41
3.5 Query result overlaid on picture............... 41
4.1 Plot of Landuse Region Class 44
4.2 Plot of Contours Region Class 45
4.3 Plot of 100 Year Flood Plain Region Class............... 46
4.4 Plot of Combined Point and Line Layers............... 47
4.5 Plot of Road Overlays 48
4.6 Plot of Rivers Class 49
4.7 Plot of Powerline Class 50
4.8 Plot of Railroads Class............... 51
4.9 Plot of Point Data Layer 52
4.10 Query Number 1 59
4.11 Query Number 2 60
4.12 Query Number 3 61
4.13 Query Number 4 62
4.14 Query Number 5 63
4.15 Query Number 6 64
4.16 Query Number 7 65

List of Tables

1	AGIS Query Language Operators and Feature Types	31
2	Sample Query Commands	32
3	Organization of Test Data by Feature, Class and Layer	53
4	File Size	54
5	File Creation	55
6	Test Queries for AGIS system	57
7	Accumulated Query Processing Time	58

CHAPTER 1

INTRODUCTION

This report covers work performed during the second year of a study of data representation and processing. in what is sometimes called a hexagonal structure. The approach which was studied has been developed at Interactive Systems Corporation (ISC) over a period of several years. It is based on a coordinate system called GBT which addresses hexagonal regions in a plane.

During the first year of the study ISC software which uses the hexagonal scheme was tested on polygonal map data such as land use areas and flood plains. Results of the project are described in Hexagonal Data Base Study (September 1983). Portions of the theoretical discussion given in that report are included in Chapter 2 of the current report.

In the second year's work on this project we have incorporated raster imagery into the system. Line and point data were integrated with the polygon information. An extended query capability was added to handle these new data forms. The resulting enhanced system was tested on a small data set supplied by the U.S. Army Engineer Topographic Laboratories (ETL).

CHAPTER 2

THEORY AND IMPLEMENTATION

2.1 Overview

The software which was created and exercised in the course of this study was built to work with Interactive Systems Corporation's commercial software for geographic information processing. This system, called AGIS, consists of an interactive graphics subsystem (IGC), a hexagon based spatial data manager (GRAM), a subsystem for generating map displays (ODM), and a relational data manager for handling non-spatial (attribute) information. The relational data manager used for the study was the Relational Information Management (RIM) system, a software package developed by the Boeing Company for NASA. Figure 2.1 illustrates the relationship between AGIS components. It should be emphasized that both GRAM and RIM are general purpose data managers. GRAM files can contain any type of spatial information, for example, pixels, drawings, or graphs. AGIS uses GRAM to build map files for storing and processing geographic information. Thus, AGIS exists as a layer of software built on top of GRAM and RIM in order to handle this map data. In the first five sections which follow, we are concerned more with general properties of GRAM and RIM used by AGIS to represent geographic information.

During the first year of the study the capability for representing and analyzing polygon data was built into AGIS. In the second year the software was enhanced to incorporate both polygon and line and point data in the same data base. In order to do this a scheme was required which would allow information about point, line, and polygon features stored in RIM files to be identified with the spatial representation of the features stored in GRAM files. It was necessary that the data be easily edited, displayed, and queried.

2.2 Nodes and Relations

The hexagonal data structure which underlies GRAM is designed to handle spatial information, that is, data for which location or position in some geometric reference system is the key attribute. The basic entity in this data structure is called a node and represents either a hexagonal region or an aggregate of hexagons. In either case, the node has a corresponding GBT address. Data coordinates in other reference systems are transformed into the GBT system. In mapping a data set into the GBT system, consideration is given to the resolution of the underlying GBT grid. For example, if no two points in the data set are closer than 50 meters, a grid built so that the distance between adjacent hexagon centers represents 50 meters or less will insure that no hexagon contains two data points. In most applications the resolution of the grid is selected to be less than the resolution of the data so that no two points will be represented by a single node in the data structure.

There are two basic relations defined on nodes which permit more complex data such as lines or polygons to be represented in the data structure. These relations are:

- connectivity
- set membership.

AGIS

AUTOMATED GEOGRAPHIC INFORMATION SYSTEM SOFTWARE MODULES

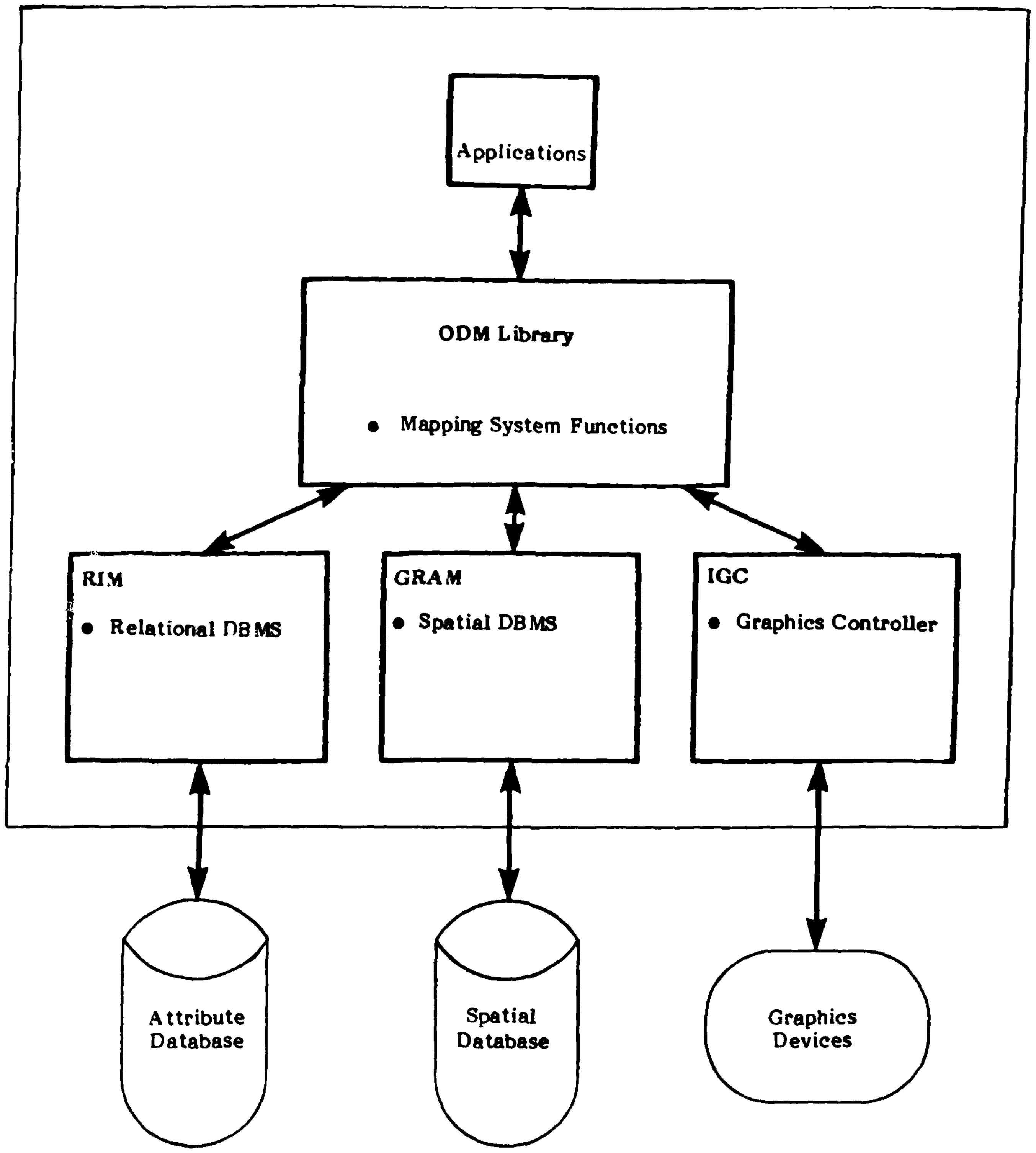


Figure 2.1 The functional relationship between the AGIS subsystems.

A node can be connected to each of any number of other nodes. The number of nodes to which it is connected is called its degree (see Figure 2.2). There are many kinds of node connectivity. The connectivity type used is a function of the implemenation. For example, connectivity can have an associated equation defining a curve between the points represented by the two nodes. This is useful in graphics applications. The connectivity used in this study, the "packed vector" form, associates a sequence of intermediate points with the node connection.

Lines and polygon boundaries are examples of node sets. In applications, other attributes may define such sets. For example, "county line", "interstate highway", and "land use area" all give rise to node sets. The nodes in these sets are connected to each other in such a way that lines or boundaries are accurately represented. Area features are defined by their boundaries. Figure 2.3 shows some node sets representing map data. An important aspect of the set membership relation is that a given node can belong to more than one set. This means that data of different types can be naturally integrated by using location. Thus, the nodes of a line which has several properties are present in the data structure only once.

2.3 The Addressing Scheme

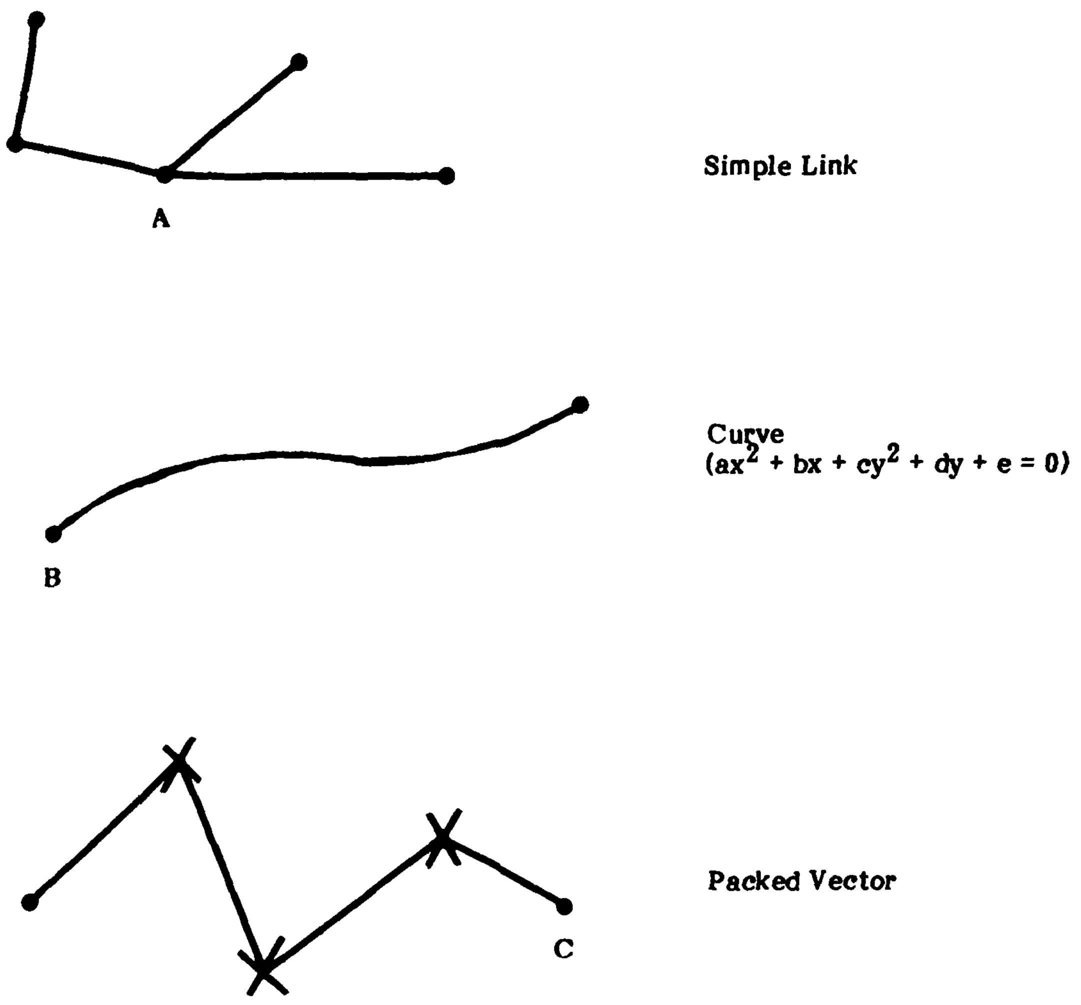
Each node in the data structure has a unique address in a coordinate system called Generalized Balanced Ternary (GBT). We have stated that the nodes correspond to hexagonal areas and to collections of hexagons.

Hexagons have certain advantages over other tessellations or tilings of the plane. The hexagonal covering has the uniform *adjacency* property, that is, each element of the covering is adjacent to exactly six other elements and shares with each exactly one-sixth of its boundary. In contrast, a covering of the plane with squares does not have uniform adjacencies. Some squares are adjacent at a point while others share a side. It is on the hexagonal cells that the GBT system is built as a hierarchy. At each level, the cells are constructed of cells from the previous level according to a rule of aggregation.

A first-level aggregate is formed by taking a single hexagon and its six neighbors (see Fig. 2.4a). The first level aggregates also cover the plane and have the uniform adjacency property. In general, an aggregate at level n is formed by taking a level n-1 aggregate and its six neighbors. It can be shown that the planar covering and uniform adjacency properties hold at each level. Figures 2.4b and 2.4c show second- and third-level aggregates.

The GBT addressing system is based on the following scheme. In an aggregate, the center cell is labeled 0 and the outer six are labeled, in clockwise order, 1, 3, 2, 6, 4, 5 (see Fig. 2.4a). Each hexagon in the plane has a unique GBT address, a sequence of digits corresponding to the labels of the cells above that hexagon.

Each digit of the address corresponds to an aggregate level. For example, the address 536 labels the hexagon in the 6 position of the first-level aggregate, which is in the 3 position of the second-level aggregate, which is in the 5 position of the third-level aggregate, which is at the 0 or center position at all higher levels. The hexagon 536 has a dot in it in Figure 2.4c.



● **Node**

X Intermediate Point

Degree of Node A = 3

Degree of Node B = 1

Degree of Node C = 1

Figure 2.2 Node Connectivity

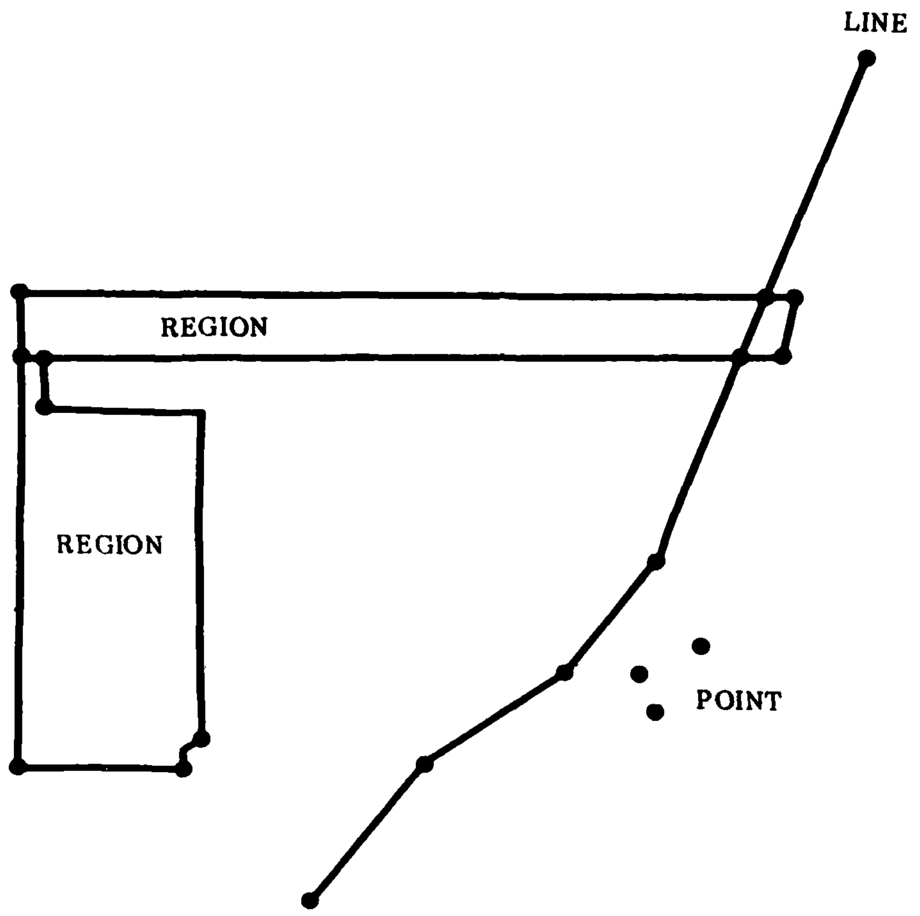
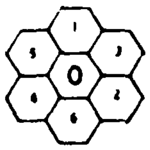
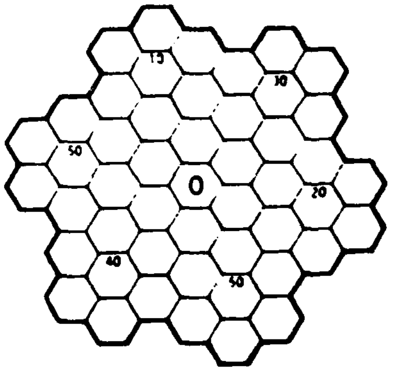


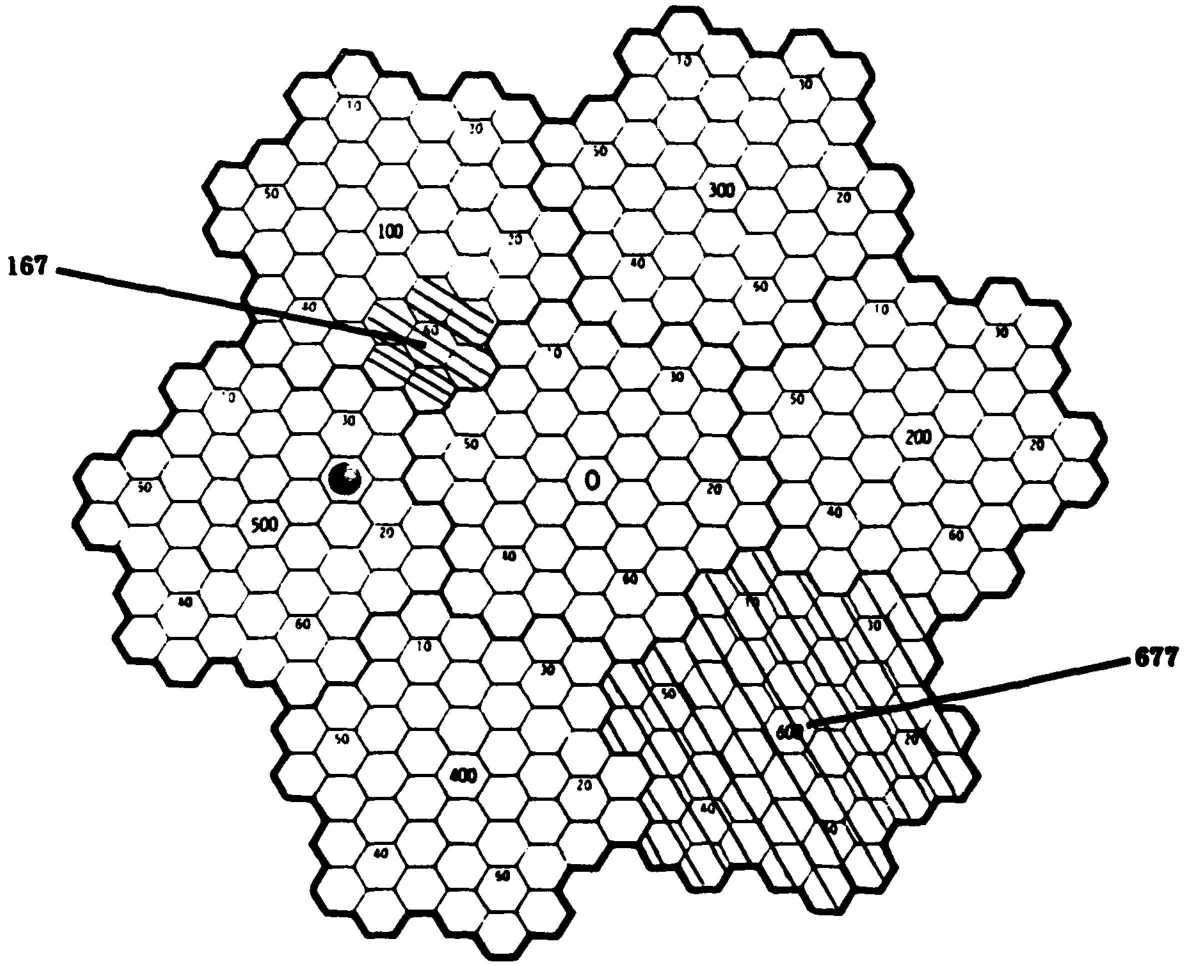
Figure 2.3 Node sets representing map data.



a. First Level Aggregate



b. Second Level Aggregate



c. Third Level Aggregate

Figure 2.4 The aggregate structure..

The digit 7 has a special meaning in a GBT address. It is used to address aggregates rather than individual cells. In Figure 2.4c, the first level aggregate 167 and the second level aggregate 677 are shown as shaded regions. By utilizing the digit 7, regions (aggregates) of various sizes can be addressed, which is a useful property in image analysis applications [1].

The GBT system has an arithmetic structure which allows the addresses to be added, subtracted, and multiplied. These operations correspond geometrically to the standard vector operations in the plane. The arithmetic is described in more detail in [2], [3], and [4].

GBT as implemented in the computer requires 3 bits per digit. The GRAM data manager permits addresses of up to 20 digits to be stored as 8 byte variables. A universe of twenty digits has 7^{20} addressable hexagons. This permits an area equal to the surface of the earth to be filled with hexagons each of a radius of approximately 5 centimeters.

This GBT scheme would be cumbersome if it were necessary to convert to other coordinates in order to perform mathematical operations. The nodal structure used for representing geographical information requires many such operations in processing the data. As implemented, the arithmetic functions are done entirely within the GBT system.

2.4 The Data Tree

The nodes, which are the basic entities of the data structure, and the relations on them exist in the computer as data records in files. The GBT addressing system permits these files to be structured as trees [5]. By this we mean that there is an order relation (a partial ordering) on the nodes. Since each node represents a hexagon or a GBT aggregate of hexagons and has a unique GBT address, the order can be defined simply: "A is higher than B" means that the area represented by B is a subset of the area represented by A. In Figure 2.5, B is the level one aggregate 327 while A is the level two aggregate 377 which contains B. This GBT order results in a tree with one highest or root node representing the universe. Each node in the tree has the potential for seven subordinate branches (see Figure 2.6).

The tree is implemented in the GRAM data manager in a way that permits it to change with the amount and distribution of data. Initially, a newly created GRAM file has only one tree node in it - the root node. As a program begins to populate the file with data records, the list of records at the root node grows until it reaches a preset maximum length.

A process called "subdividing" is then performed to break the list of records into smaller lists, and then the appropriate tree nodes are created at the next level. The subdividing process occurs whenever the list of records at any given tree node exceeds the maximum allowed.

This technique is efficient in several ways. First, no tree nodes or other structure exist to support areas of space in which there is no data. Second, the refinement or depth of the tree varies as a function of the density of the data in a particular area. In areas where the data is dense, the tree will be full while sparse areas will yield an abbreviated structure. Third, the processes of subdividing and of pruning back the tree take place automatically and continuously as data records are added or deleted.

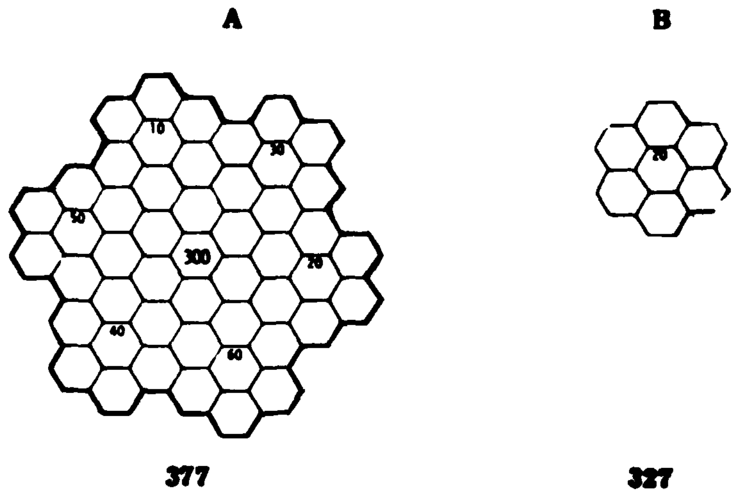


Figure 2.5 The aggregate 377 is higher than 327.

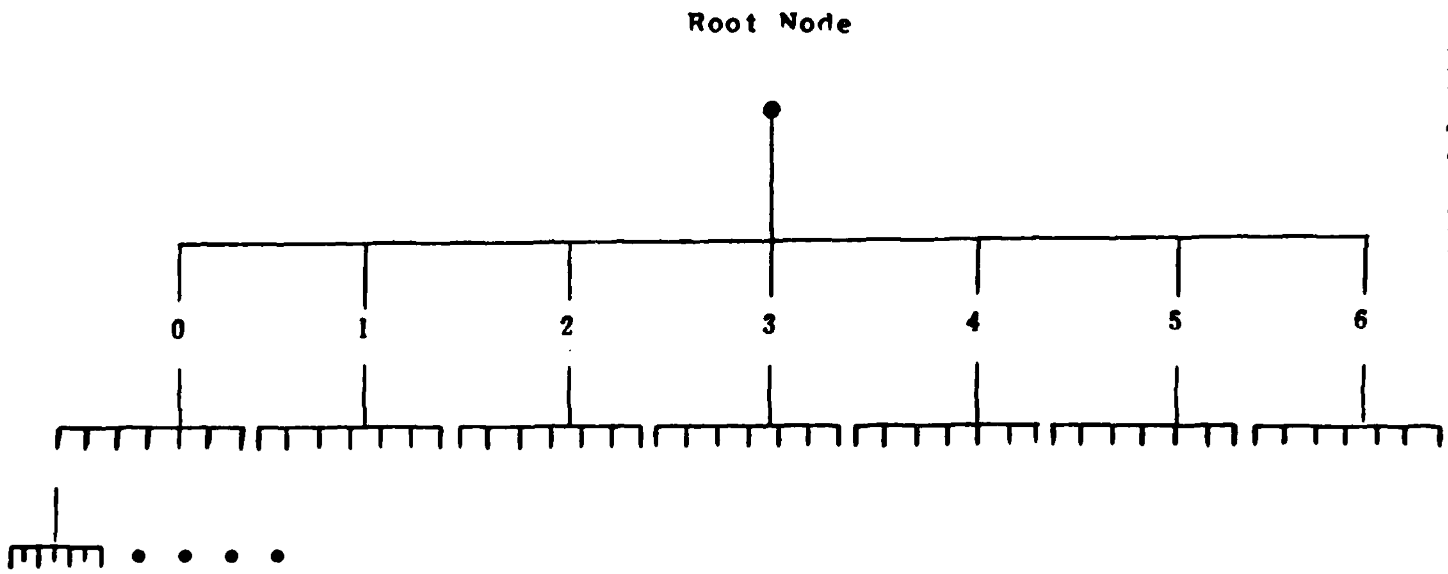


Figure 2.6 The tree generated by a set of GBT addresses.

Finding data records by location with the tree works like this. To find the data record located at address 5133 in the GRAM tree depicted in Figure 2.7, begin at the root node. This has four subordinates, one of which is "5", the highest digit in the address. At the node "5" there are three subordinates, one of which matches the second highest digit "1". Moving to that node, three subordinates are found. The subordinate node "3" corresponds to the third highest digit "3" in the address. This node is a terminal one. Any lower records will be present in a numerically ordered list stored at the location. The address 5133 is on that list. The data record for 5133 has been found in this example by passing through four tree nodes and examining a numerically ordered list. Since the length of this list is a database parameter set by the user, the desired tradeoff between list processing and tree processing can be made.

2.5 Searches

The GBT data tree permits access to the nodes by their addresses, (which correspond to locations in space), in the manner described in the previous section. In order to search an area (either circular or polygonal) for data, a list of aggregates is generated to provide a covering that exceeds the original search area (see Figure 2.8).

The associated tree nodes are then examined for data records and candidate data is checked against the original area. These search-by-location functions are part of the GRAM software. In addition GRAM has some capability for searches on non-locational attributes. Figure 2.9a shows the general form of a tree node data record. A portion of this record, the Boolean field, can be used to store state or characteristics information in individual bits. Figure 2.9b gives an example in which one portion of the field indicates node data type and another portion gives the node degree.

At each node in the tree, a composite Boolean field is automatically maintained. This composite field represents the logical union of the Boolean fields of all the data records that reside in the area represented by that tree node. The root node contains the composite Boolean field of all data records in the file, and lower nodes contain composite Boolean fields of successively smaller areas of the file. A terminating tree node contains the composite field for only the data records that reside at that node. These composite fields are continuously updated by GRAM as data records are added, modified, or deleted.

GRAM is able to retrieve data records based on the Boolean field. Any of a set of relational and logical tests can be applied against specific bytes or words of the Boolean field in the records. The node data records also contain secondary characteristics, such as attribute information in user specified format. GRAM does not perform file searches based on this data.

2.6 Attribute Data Management

GRAM itself has limited attribute data handling capabilities. Such information about nodes as their degree and whether they are part of lines or boundaries can be encoded in the GRAM record. The more complicated data associated with geographic entities is more appropriately handled by a relational data manager. For this application, the AGIS system uses the Relational Information Manager, RIM.

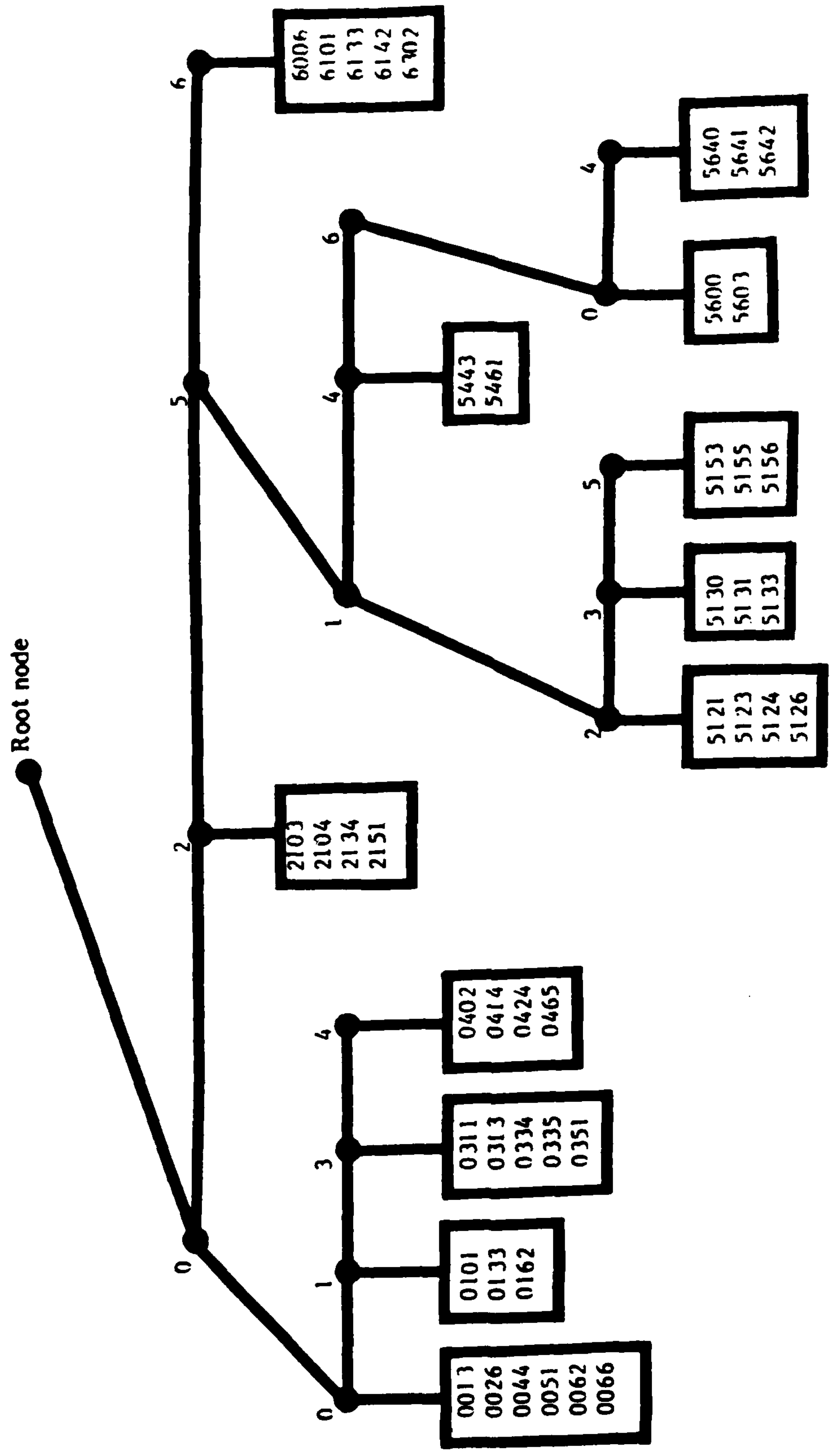


Figure 2.7 A GRAM Tree

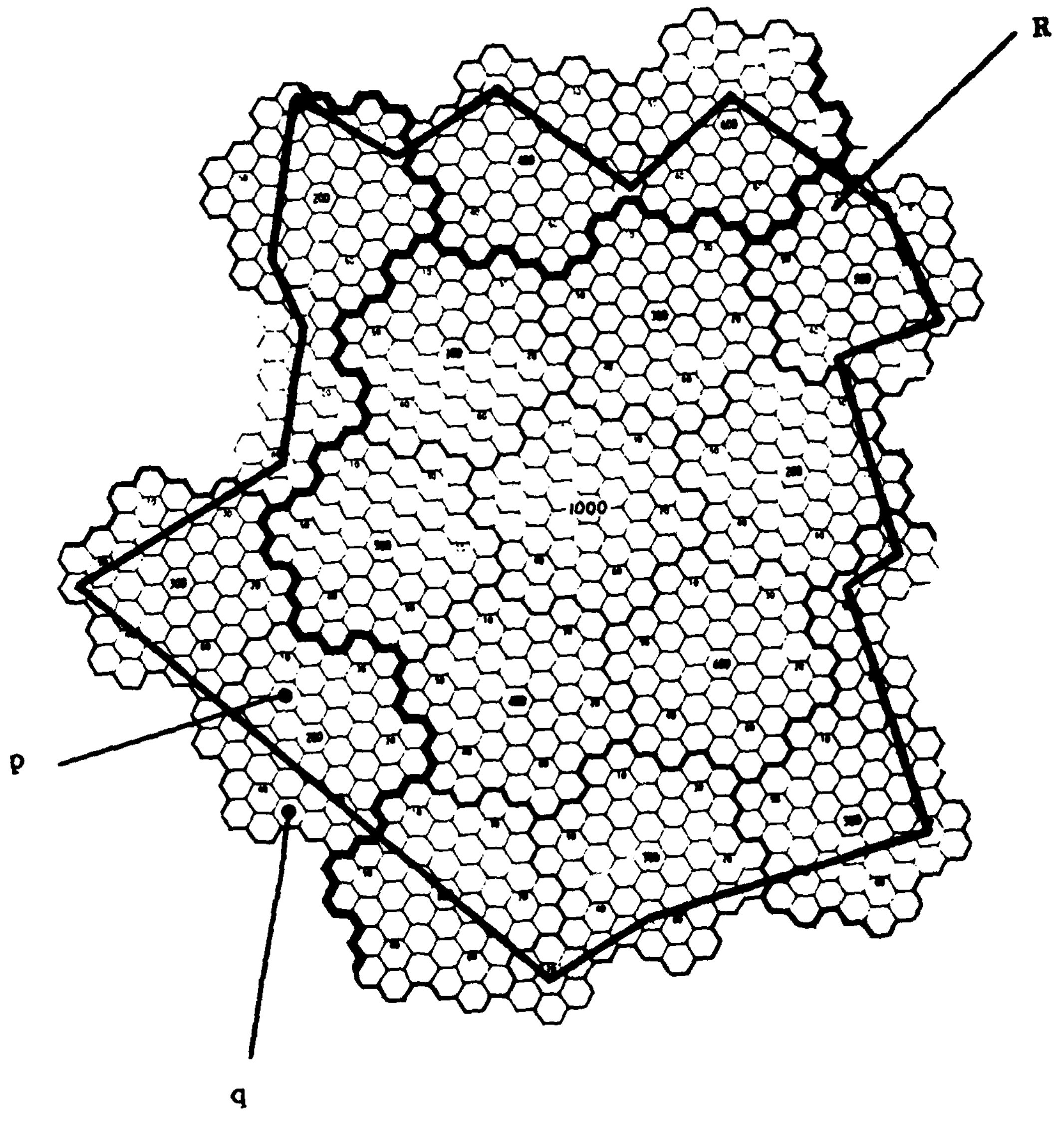


Figure 2.8 Area Search. Level one, two, and three aggregates which form a cover of the region R (indicated by its polygonal boundary). Retrieval of data by aggregate will return both points p and q. Further testing eliminates q which is outside of R.

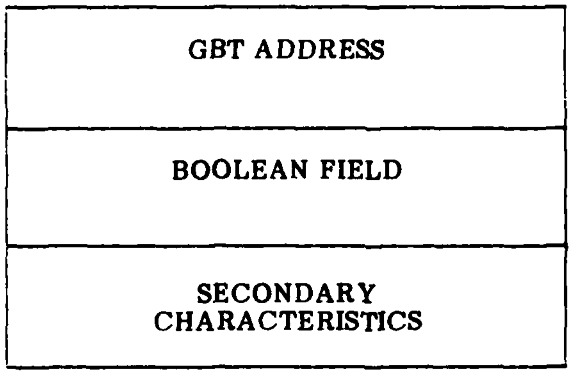


Figure 2.9a General tree node data record.

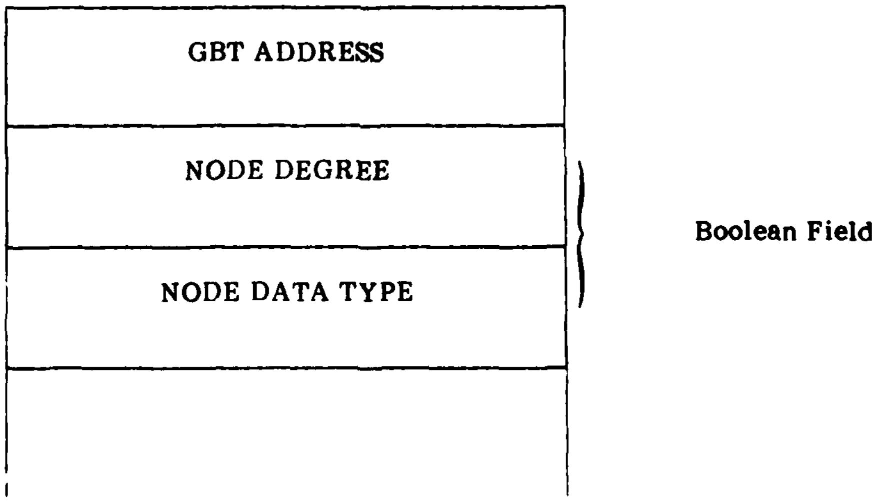


Figure 2.9b Tree node with Boolean field used for node data type and degree.

A RIM data base consists of relations. A relation is a collection of records, each made up of a tuple. The terms of the tuple are called attributes. A RIM command requests either all or some attributes from those records which satisfy a set of conditions called a "where clause". "Where clauses" use relational operators such as "equal to", "less than", or "not equal to". This RIM command:

```
SELECT ALL FROM REL WHERE FIELD1 EQ LANDUSE AND FIELD2 EQ POLYGON
```

will result in all complete tuples being retrieved from the relation REL which have FIELD1 element equal LANDUSE and FIELD2 element equal POLYGON.

2.7 AGIS - Linking RIM and GRAM

AGIS makes use of RIM and GRAM to store geographic information. AGIS allows the user some flexibility in how the information is stored but for the most part this is done automatically by AGIS and so is out of the user's control. In the next chapter the user interface with AGIS is discussed in more detail. In the remainder of this chapter a description of how AGIS represents the specific types of geographic entities which are part of this study is given. These entities are points, lines, and regions (polygon interiors). The term map file is used interchangeably with AGIS file throughout. It should be noted that these are specially structured GRAM files.

In order to represent the three types of spatial features, AGIS uses data records called signposts. Created by AGIS when the user assigns attributes, signposts are the means by which AGIS links spatial data and attribute information. All the signposts for a map file are stored together in a separate GRAM file. This is done in order to take full advantage of GRAM's speed in searching the data tree based on attributes and location. The signpost file for a typical map database is much smaller than the full map file.

There are five types of signposts, each containing this information:

- location
- feature type
- data class
- signpost type
- direction vector
- subject identification

For each signpost, its location corresponds to a node that belongs to the spatial feature. Feature type is point, line, or region. Features are organized into classes at the user's discretion. The signpost type is set according to the feature type it is associated with. The direction vector is used to define lines and regions as described below. Each signpost has a subject identification which is a user assigned text string description (e.g. street name, elevation, etc.).

Signposts serve as a link between the map file and the RIM file. They are linked by location to the spatial data stored in the map file. A signpost will share its location with a node in the map file. All the attribute fields of the signpost will also appear as data elements in a RIM relation file. In this way the signpost is linked to a RIM tuple.

Figure 2.10 shows a network of nodes that might be stored in an AGIS map file. Each dot in the figure represents a node, which has a corresponding GBT address. The lines that connect the nodes represent the links between file records. The circled nodes have signposts associated. These are described in more detail below.

2.8 Point Data

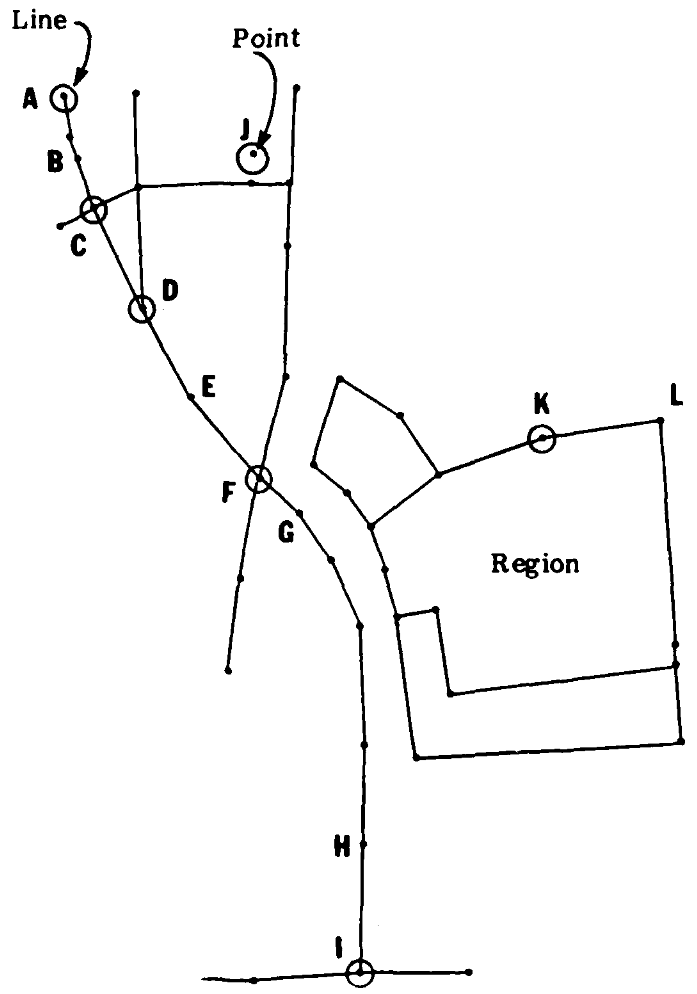
Point data in AGIS is represented by a single node in the map file, a signpost, and a RIM record. In Figure 2.10, node J is a single point. Figure 2.11 shows the signpost for J and the data elements in the RIM tuple associated with J. Given the location of a point, the corresponding signpost can be retrieved from the GRAM file. The attributes stored in the signpost (feature type, class, layer, direction, and subject identification) are used by the system to build a RIM interface command. RIM then retrieves all other attributes associated with this point.

2.9 Line Data

There are three kinds of signposts used in defining line features: start, end, and junction signposts. In Figure 2.10, there is a line going from A to I. It crosses other lines at C, D, and F. Therefore, in AGIS this linear feature would have five signposts: a start signpost at A, an end signpost at I, and three junction signposts, at C, D, and F. Figure 2.12 shows the start, end, and one of the junction signposts. Notice the use of the direction vector. For the line start it points the way out of A to the end of the line. At each junction, it tells which path out of the node leads to the end. At the line end, I, it points back to the start. A line feature has only one RIM tuple. Its direction vector is the same as the start signpost. In this way, beginning with the RIM tuple for a line (retrieval based on attributes) the line can be traced in the map file from start to end.

2.10 Region Data

A region feature has only one associated signpost which is located at a node on its boundary. In Figure 2.10, for the region indicated, the signpost is at K. The selection of signpost locations for line and point data is strictly defined. For a region, AGIS can associate the signpost at any boundary node. The choice is arbitrary. Figure 2.13 presents a sample signpost for the region in Figure 2.10. The RIM tuple for that region is also shown. The direction field in the signpost is shared by the tuple and is essential in defining the region. Regions are traced using the right-hand rule which says at any node of degree greater than two, always take the rightmost path out of the node with respect to the entrance path. A rightmost path can be computed easily in an ongoing trace operation given the list of vectors from a node. However, if we start at a node on a region boundary, the direction or path out of the node which must be taken in order to trace the boundary cannot be determined until the direction vector has been retrieved. For this reason the region signpost has a direction field. In Figure 2.10, the region is traced by going from K to L and taking the rightmost path at each node of degree greater than two.



Nodes and Connections in a Map File

Figure 2.10 The node network is stored in a GRAM file.

POINT SIGNPOST

	SIGNPOST	**RIM TUPLE**
Location:	J	J
Feature:	Point	Point
Class:	Church	Church
Layer:	Point Data	Point Data
Direction		
From:	J	J
To:	J	J
Subject ID:	"St. James Church"	"St. James Church"
		User Defined

Figure 2.11 Point Signpost. The signpost for node J is stored in a GRAM file. A RIM record is also stored.

LINE SIGNPOSTS

	SIGNPOSTS			RIM TUPLE
Type	**Start**	**End**	**Junction**	
Location	A	I	C	A
Feature	Line	Line	Line	Line
Class	Highway	Highway	Highway	Highway
Layer	Lines	Lines	Lines	Lines
Direction From:	A	I	C	A
To:	B	H	D	B
Subject ID:	"Redwood Highway"	"Redwood Highway"	"Redwood Highway"	"Redwood Highway"

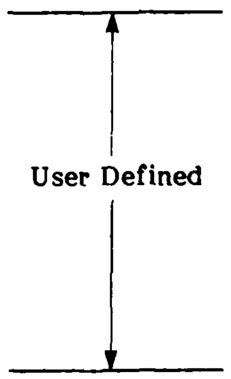


Figure 2.12 Line Signposts. A line will have starting, ending, and intermediate signposts It has only one RIM tuple.

REGION SIGNPOSTS

	SIGNPOST	**RIM TUPLE**
Location:	K	K
Feature:	Region	Region
Class:	Landuse	Landuse
Layer:	Region Data	Region Data
Direction		
From:	K	K
To:	L	L
Subject ID:	"AAV"	"AAV"

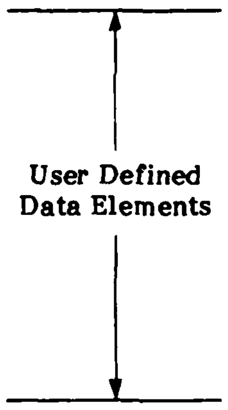


Figure 2.13 Region Signposts. A polygon has one signpost and one RIM tuple.

Figure 2.14 illustrates the use of region signposts. Suppose that a user selects the point A inside the region R_1. A search is made of the map file to find the closest region boundary node B. Starting at this node, the boundary is traced by taking the rightmost path (with respect to the vector AB) leaving B. At each node of degree greater than two, the rightmost path is taken. Eventually the path will return to B. At some point a signpost node will be reached. If the direction vector in the signpost corresponds to the path being taken, then this is the signpost for the region. As with points and lines all associated attribute information can be retrieved from the RIM file using the feature type, class, layer, direction, and subject identification stored in the signpost. The example shown in Figure 2.14 has two signposts, C associated with R_1 and D associated with R_2. In tracing the boundary of R_1, the signpost D would be rejected since its direction vector does not point in the direction of the trace.

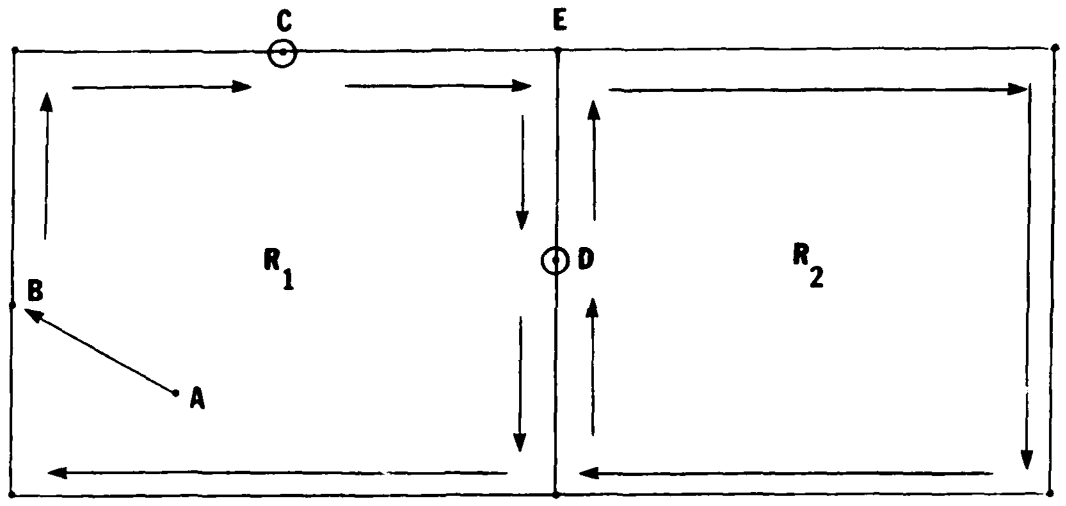
2.11 Topology Verification

It is often the case that map regions do not correspond to the interiors of polygons or simple closed curves. It is not uncommon for a map region to be bounded by two or more disconnected polygons. Figure 2.15 shows examples of such regions. We have just described how a region is defined in the node network using the righthand rule to trace its boundary. Clearly, if a region has disconnected boundaries, this algorithm will fail. In AGIS this special case is handled by connecting such boundaries with a link called an isthmus. In Figure 2.16a the inner and outer boundaries of the region R are connected by an isthmus from node A to node B. In Figure 2.16b two isthmuses are used. Notice that it is now possible to trace the entire boundary using the righthand rule if we trace each isthmus once in each direction.

One of the tasks in the current project was to develop and implement an algorithm which automatically inserts isthmus links in a node network. Such an algorithm eliminates the need for an operator to associate region boundaries manually. This function was incorporated as part of a more general program to verify the topology of a region map class. Other checks are made for the presence of one degree nodes and for lines which cross but whose intersection is not present as a node in the file. Figure 2.17 shows a region overlay which contains examples of these anomolies.

The check for one degree nodes is simple in the AGIS map file. The node degree is stored in the bitmap for each node in the map file. It can be quickly examined with a standard GRAM search.

Checking for unmarked line crossings is not as straightforward. The line crossing algorithm uses a file parameter which is given at the time the map file is created. This is the maximum vector length , d. No two nodes in the file which are farther apart than d can be linked directly to each other. If they are more than d apart some intermediate node on the line segment between them will be present in the file. Suppose A and B are linked nodes in a map file as shown in Figure 2.18. In order to check that no line crosses the segment AB the algorithm searches a distance of $\sqrt{2}d$ around A and B. If a node C is found, the algorithm computes, for each node D linked to C, the intersection, if any, of the line AB and the line CD. If there is one, between A and B, it is automatically added to the map file. The maximum vector length parameter limits the search area and therefore the number of node pairs to be considered while insuring that any intersecting segments will be found.



REGION SIGNPOSTS

Location:	C	D
Feature:	region	region
Class:	class 1	class 1
Layer:	sample	sample
Direction		
From:	C	D
To:	E	E
Subject ID:	"R_1"	"R_2"

Figure 2.14 The right-hand rule orients region boundaries and permits the correct signpost to be identified.

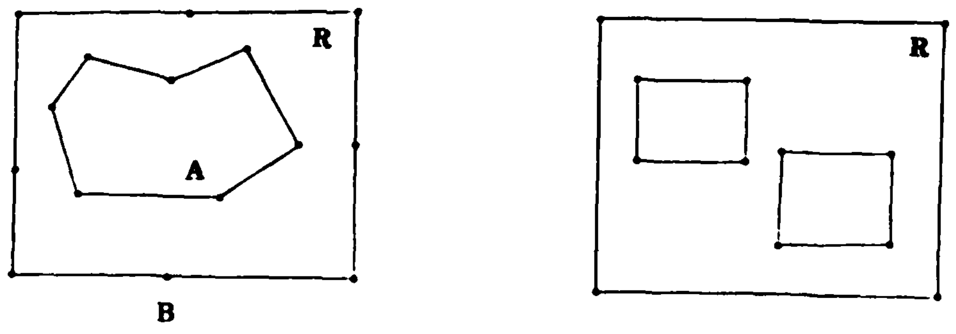
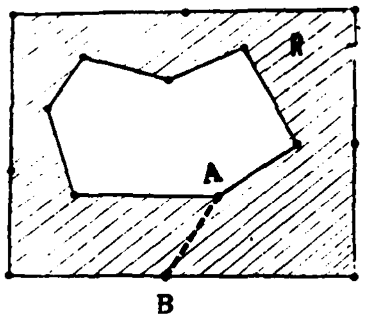
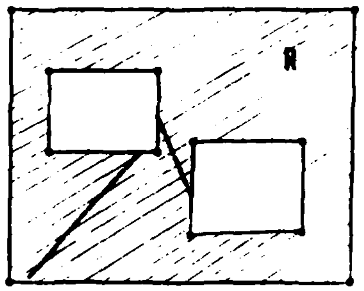


Figure 2.15 Regions with Disconnected Boundaries.



a.



b.

---- Isthmus

Figure 2.16 Isthmuses connect disconnected boundaries.

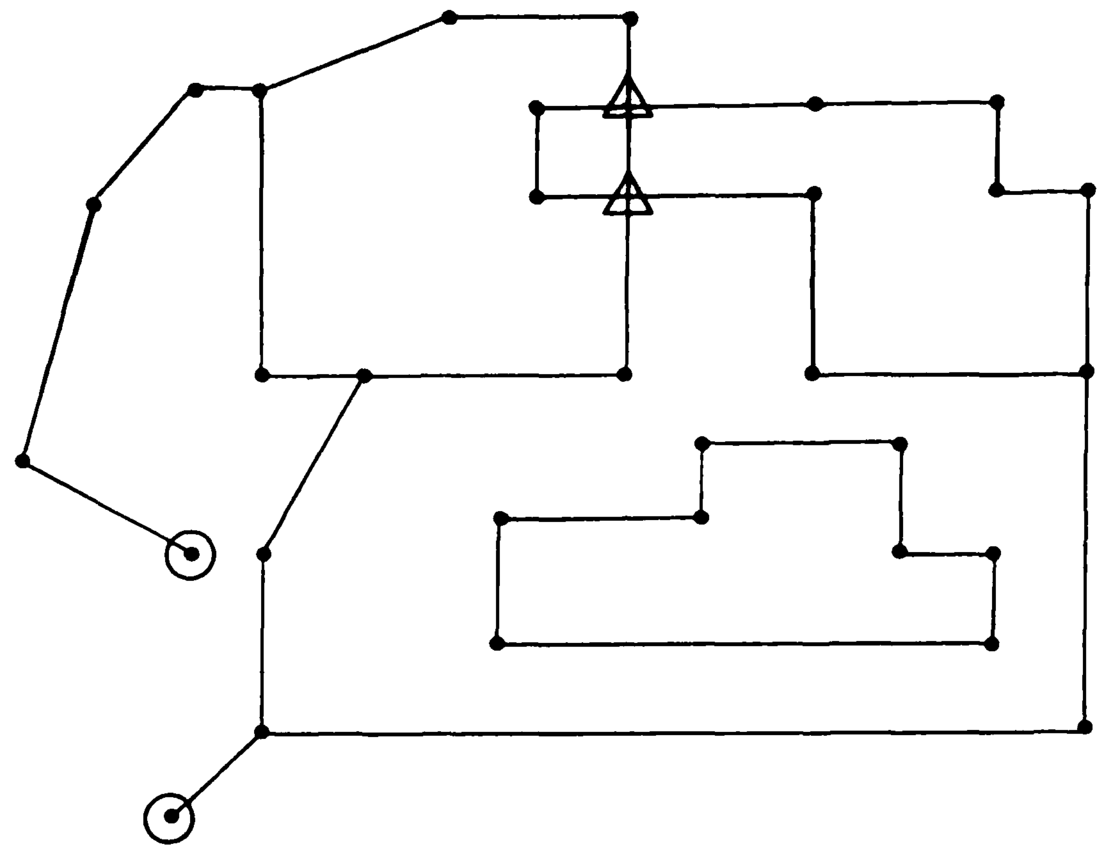


Figure 2.17 A region layer that is not topologically correct. Anomolies include an unassociated island.

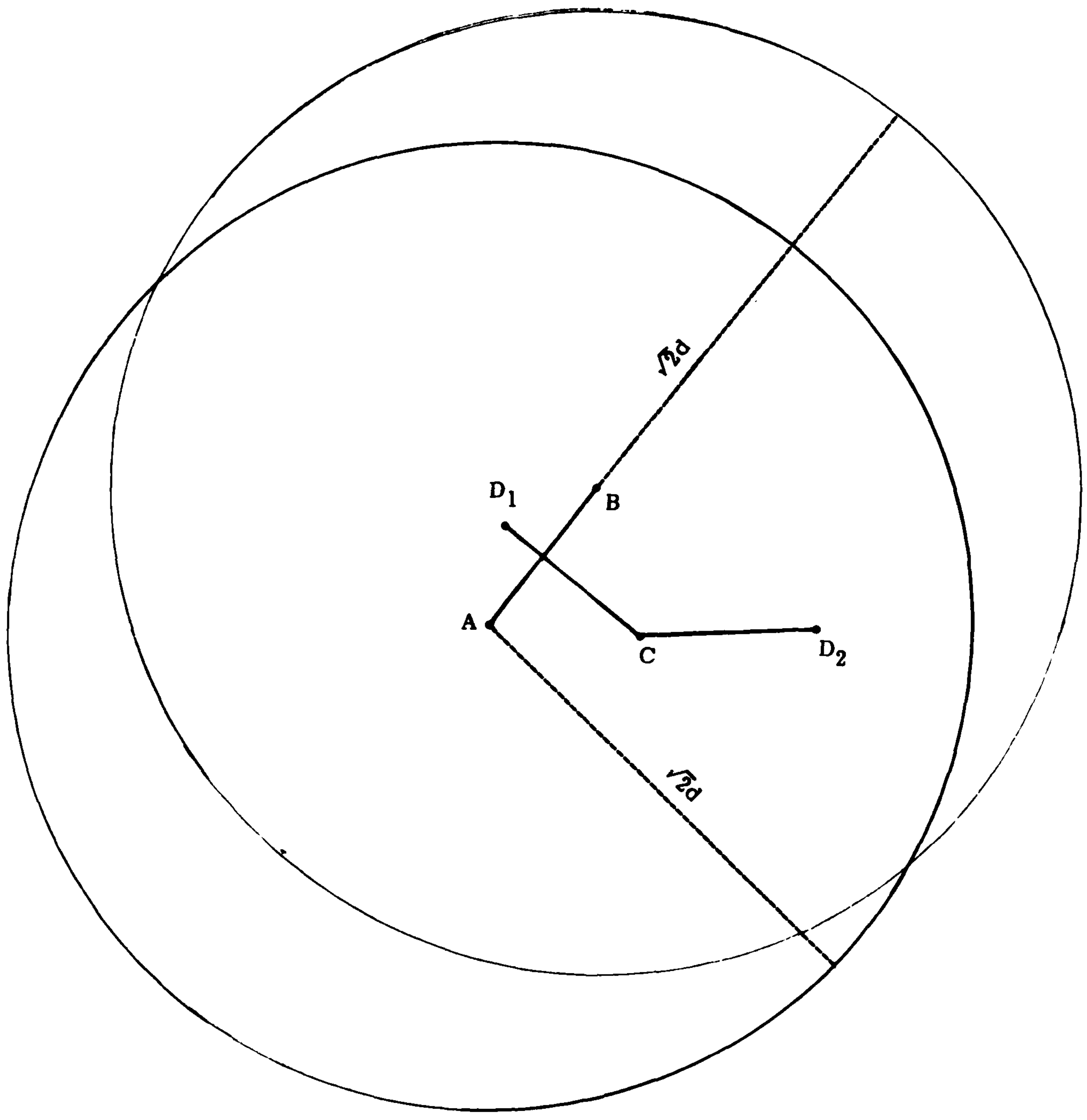


Figure 2.18 Line Crossing Algorithm checks to see if CD_1 or CD_2 cross AB.

The last step in topology verification is the automatic generation of isthmuses. The algorithm works in this way. First the node network is completely traced. The number of connected pieces or components is counted. Each component is marked and coordinates of its highest, lowest, leftmost, and rightmost nodes are recorded. Figure 2.19 shows a network with four components in which these extreme nodes are circled. The object of the isthmus generating algorithm is to link all components into one connected piece. This is done by adding isthmuses one at a time so that each new link connects two components into one new one. An isthmus should never connect a component to itself, that is, never intersect a component in two points. The algorithm is designed so that it generates isthmuses with those two properties.

If A is a component, its extreme points define a rectangle that contains A. If B is a second component, it too is contained in a rectangle. The two rectangles will overlap, nest, or be disjoint (see Figure 2.20). In each case it is true that there is at least one extreme point x on one of the two components, say A, so that a line can be drawn from x to B without recrossing A. The line may intersect a third component C, in which case the algorithm builds the isthmus from A to C along that line (see Figure 2.21). A and C are then connected, decreasing the total number of components by one. The extreme points for the combined A and C component are easily determined. The algorithm repeats this process, connecting the combined A and C to B. When all components are linked by isthmuses, a frame is built around the map area and connected to the node network at one point with an isthmus. Figure 2.22 shows the result of this algorithm on the network in Figure 2.19.

2.12 Operations on Map Layers

The effectiveness of a representation for map data can be measured by how easily and efficiently the information can be integrated and queried. For example, in AGIS as in many other data structures the information is organized into map layers such as landuse areas, roads, or landmarks. A good representation must allow the data to be combined to answer queries like "Find all roads that lie in landuse regions of type AVV" or "show all dams above 300 feet of elevation". In order to perform this sort of analysis on data stored in an AGIS database, a query language was designed and implemented during this study.

The AGIS query language uses commands which have three parts: command type, operands, and operators. Each command begins with a command type. A command then has at least one operand, which describes a set of map entities. Multiple operands are separated by operators. There are three command types. "DISPLAY" allows the resultant data to be displayed on the graphics screen. A "SHOW" command highlights the resultant data on the graphics display. The "OUTPUT" command builds a new file in which the requested data is stored.

Operands can be categorized according to their feature type, that is, as points, lines, or regions. Each operand must be formatted in this way:

"LAYER"<CLASS>FEATURE/QUALIFIER.

A qualifier is a RIM type "Where Clause".

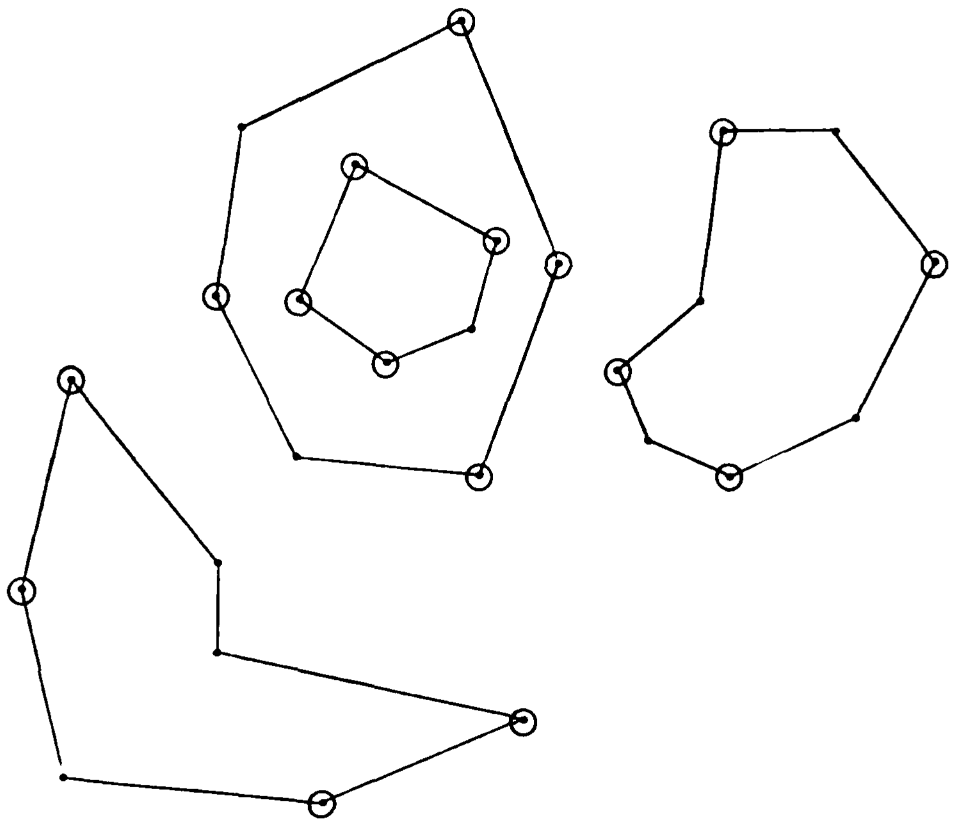


Figure 2.19 A Network with Four Components.

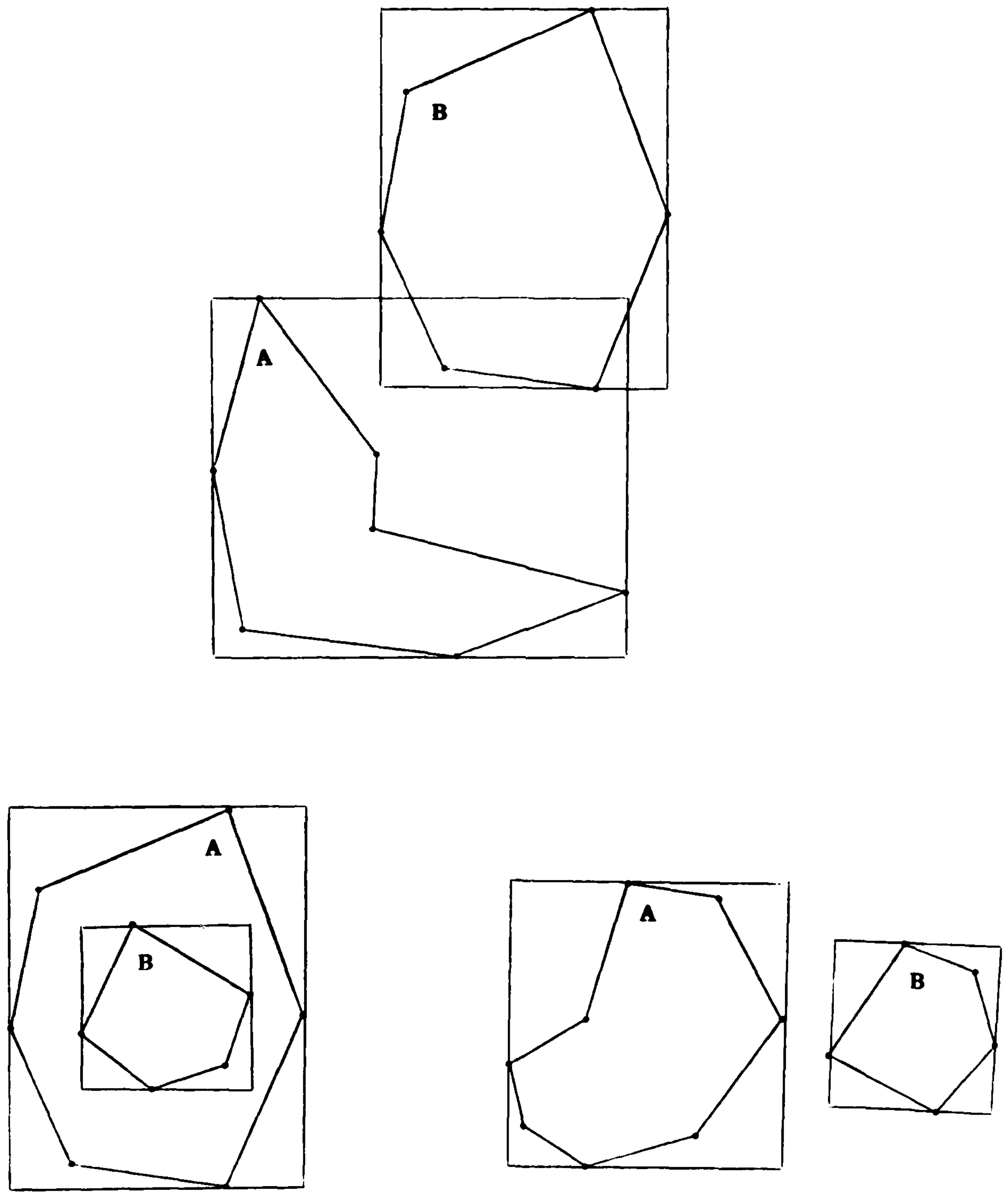
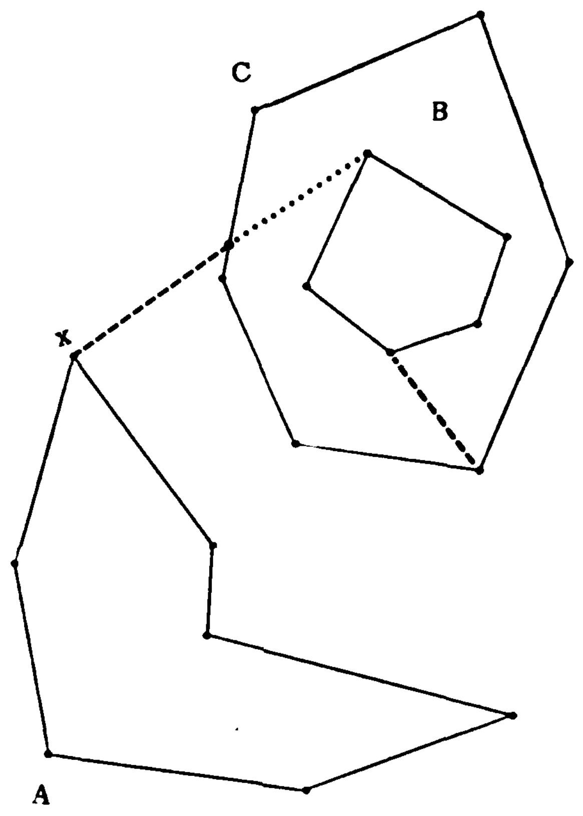


Figure 2.20 Bounding rectangles overlap, nest, or are disjoint.



- - -	Isthmus
.....	Isthmus not drawn

Figure 2.21 Building an Isthmus from Component A.

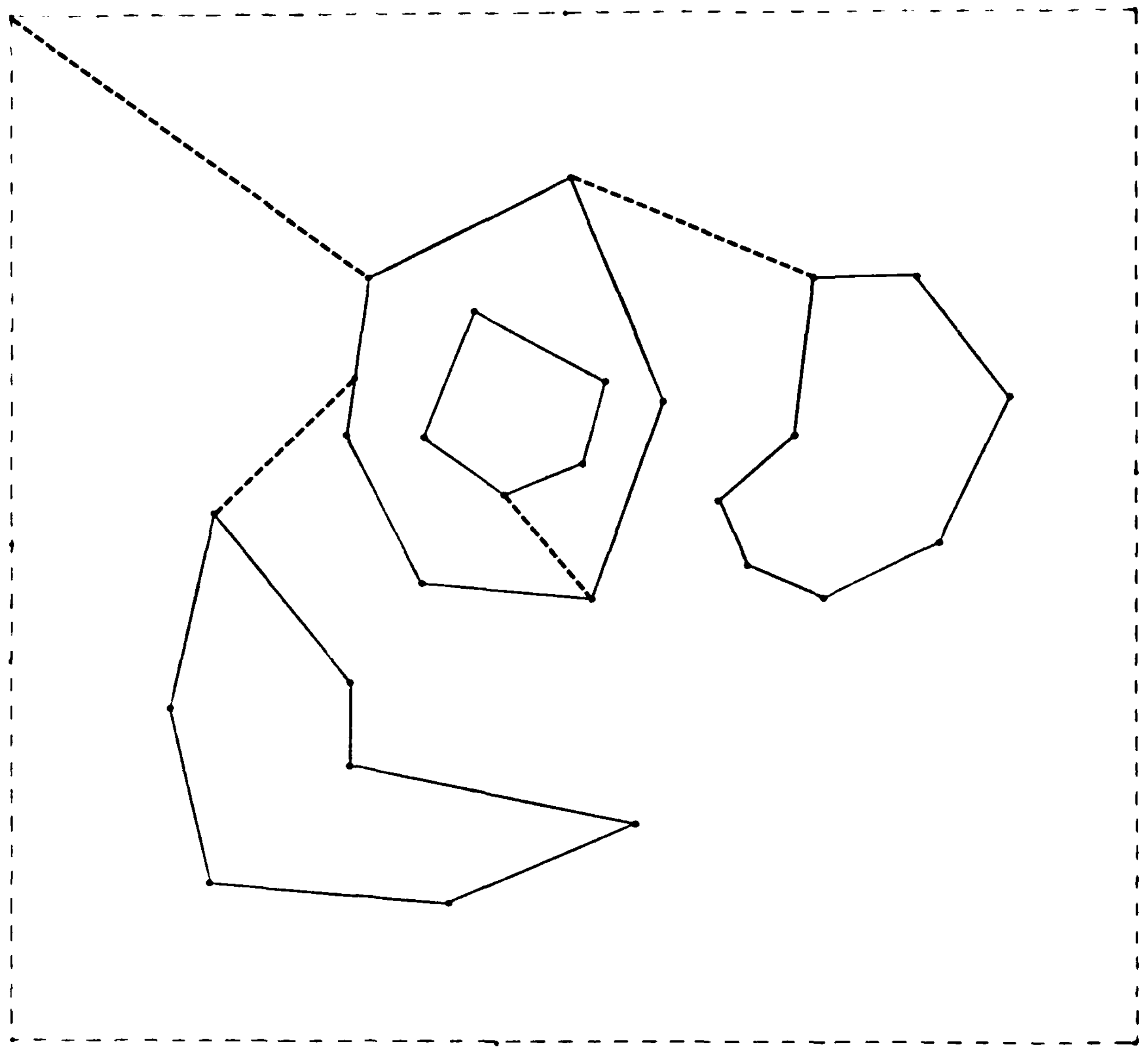


Figure 2.22 A Node Network with Isthmuses and a Frame.

The language has six operators: "and", "or", "not", "interior", "exterior", and "plus". Table 1 shows which combinations of features and operators are permitted.

Type	Result
REGION1.AND.REGION2	All areas common to Region 1 and Region 2.
REGION1.OR.REGION2	All areas that are either Region 1 or Region 2.
REGION1.NOT.REGION2*	All areas that are part of Region 1 but not Region 2.
POINTS.INTERIOR.REGIONS	All points inside the Regions.
POINTS.EXTERIOR.REGIONS	All points outside the Regions.
LINES.INTERIOR.REGIONS	All lines inside the Regions.
LINES.EXTERIOR.REGIONS	All lines outside the Regions.
ALLTYPES.PLUS.ALLTYPES	The plus operation is used to concatenate separate queries.

* Note: The "NOT" operator is an implied ".AND. .NOT."

Table 1. AGIS Query Language Operators and Feature Types

Table 2 gives some examples of AGIS query commands.

1.) Show all Churches inside the Landuse Areas with a code of "AVV".

```
SHOW "POINTS"<ALL POINTS>POINTS/WHERE[SID EQS "CHURCH"]+
        .INTERIOR."REGIONS"<LANDUSE AREAS>+
        REGIONS/WHERE[SID EQ "AVV"]
```

2.) Show all Areas in the Flood Plain that are at the elevation of less than 200'.

```
SHOW "REGIONS"<CONTOURS> REGIONS/WHERE +
        [SID LE "200"].AND."REGIONS"<FLOOD PLAIN>REGIONS/WHERE +
        [SID EQ FLOOD PLAIN]
```

3.) Show all Roads that are at an elevation of 0' to 100'.

```
SHOW "LINES"<ROADS>LINES.INTERIOR."REGIONS<CONTOURS> +
        REGIONS/WHERE[SID LT "100" AND SID GT "0"]
```

Table 2. Sample Query Commands

After a query command is parsed, the data associated with each operand is retrieved from the map file and placed with the other operand data in a temporary map file. As the data is transferred to the temporary file an "Operand Bitmap" is stored at each node. This Bitmap specifies which operand the node is associated with. For instance, in the query:

```
SHOW"LAYER#1"<CLASS1>REGIONS.AND.
     "LAYER#2"<CLASS2>REGIONS
```

All data associated with Layer 1, Class 1 would have the first bit set in the operand bitmap, and all data associated with Layer 2, Class 2 would have the second bit set.

The combinations of operand types are basically: regions and regions, regions and lines, and regions and points. Each of these is discussed separately in the sections which follow.

2.12.1 Operations on Regions and Regions

In order to describe the algorithm which is used in combining different Region Layers, the following query will be analyzed (see Figure 2.23):

```
SHOW"LAYER#1"<SQUARE>REGIONS.AND.
     "LAYER#2"<TRIANGLE>REGIONS
```

(i.e. Show all regions that are both *triangle and square.)*

As outlined in the previous section, an "Operand Bitmap" has been associated with each node in the temporary file. In this case, nodes 1 - 4 have the first bit set (10) and nodes 5 - 7 have the second bit set (01). Note that nodes 8 and 9 are newly created intersection nodes and do not have an associated operand bitmap. The regions that satisfy the query will have a query bitmap of [11] (i.e. both square and triangle).

Each face can be labeled with its query bitmap as follows:

An individual operand is equivalent to a decomposition of an area into regions with essentially a binary property: each region either possesses or does not possess the property specified by the operand. Given this property, a "Query Bitmap" can be generated for each region in the temporary file which specifies whether the region possesses or does not possess the properties of each operand in the query. If each operand is assigned a bit position in the query bitmap (first operand - first bit, second operand - second bit, etc.) then a query bitmap can be associated with each region in the file showing which operand(s) the region satisfies.

The algorithm which assigns a query bitmap to each region in the temporary file is described as follows:

1) Start outside of everything and trace the Exterior Region (Query Bitmap = [00])

2) As the Exterior Region is traced the rightmost vector is again taken at all higher degree nodes (in the example, starting at Node 1 the Region will be traced 1-4-9-7-6-5-8-2-1).

3) At higher degree nodes (Nodes 8 & 9) a query bitmap for bordering regions is put on a stack to be processed later along with a starting "seed" vector. The query bitmap is determined by doing an "Exclusive Or" operation between a query bitmap and the operand bitmap of the line being crossed. In the example, at Node 9, Vector 9-3 will be stacked with the query bitmap [01] (Exclusive or between [00] (current query bitmap) and (01) (Vector 9-7 operand bitmap), and vector 9-8 will be stacked with [11] (Exclusive or between [01] (previous step query bitmap from region 9-7-8-5-6-7-9) and (10) (Vector 9-3 operand bitmap). The same process is then done at Node 8.

4) Once all Regions have been traced and a query bitmap has been associated with the regions, all regions tagged with the answer bitmap [11] are then output.

2.12.2 Operations on Regions and Lines

The region and line queries permit lines of a specified type to be clipped to the interiors or exteriors of regions. The regions may themselves be the product of operations of the types shown in Table 1. In a query of this kind, all region operands and operators are processed first. The lines are clipped to the resulting regions.

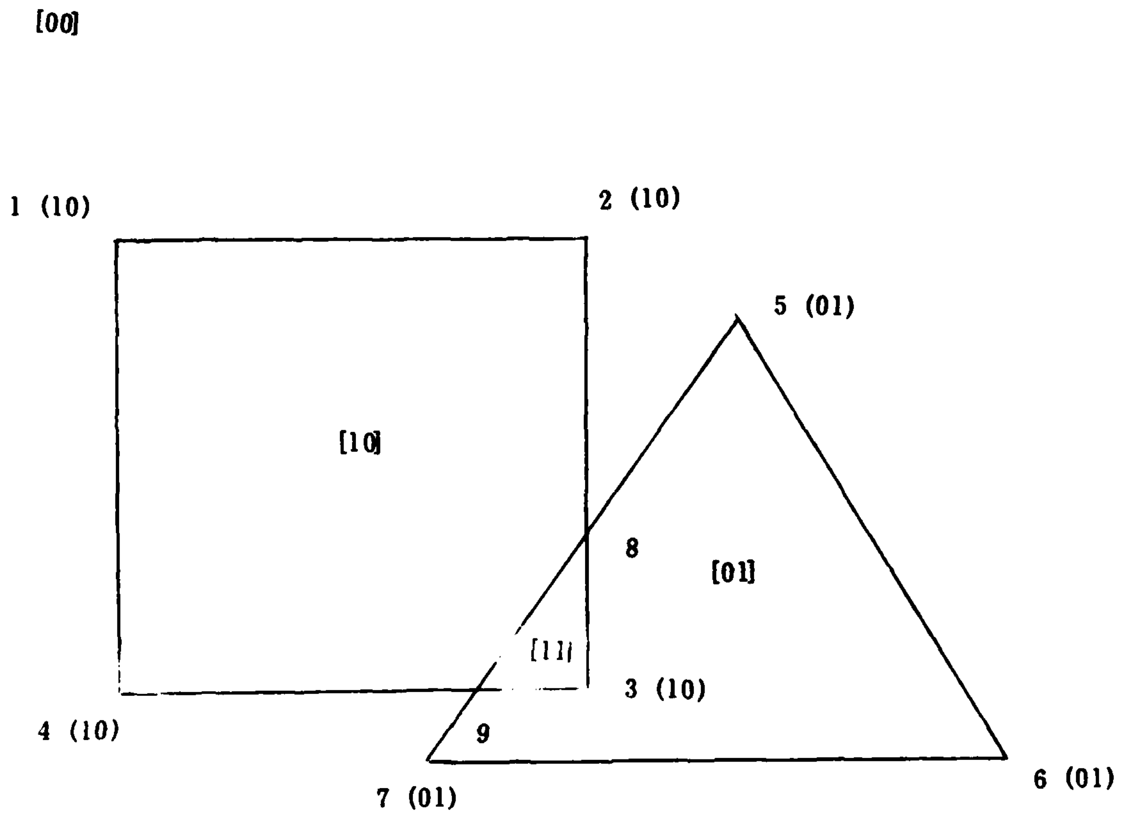
The line clipping algorithm examines each region one at a time. For a region R, each line of the correct type is checked against R. This requires that the line be traced from start to end and for each node along the line, a determination be made as to whether the node is inside or outside the region. Points of intersection of the line with the region boundary are computed. Figure 2.24 shows an example of line clipping.

2.12.3 Operations on Regions and Points

As with lines, AGIS allows the operator to find point data of a specific type which lies either interior to or exterior to regions. These regions may be specified directly or may be the result of a combination of region operands and operators.

Each region (or region exterior) is traced and then covered with a collection of GBT aggregates. Each aggregate is searched for the presence of point data. If there are points stored within an aggregate, they are retrieved and examined to see if they lie inside or outside the boundary. Those which are correctly located are placed in the temporary map file.

This type of query makes especially good use of the AGIS tree search capability and is relatively fast to perform.



(First Bit Second Bit) Operand Bitmap

[First Bit Second Bit] Query Bitmap

Figure 2.23 Query Bitmap Generation

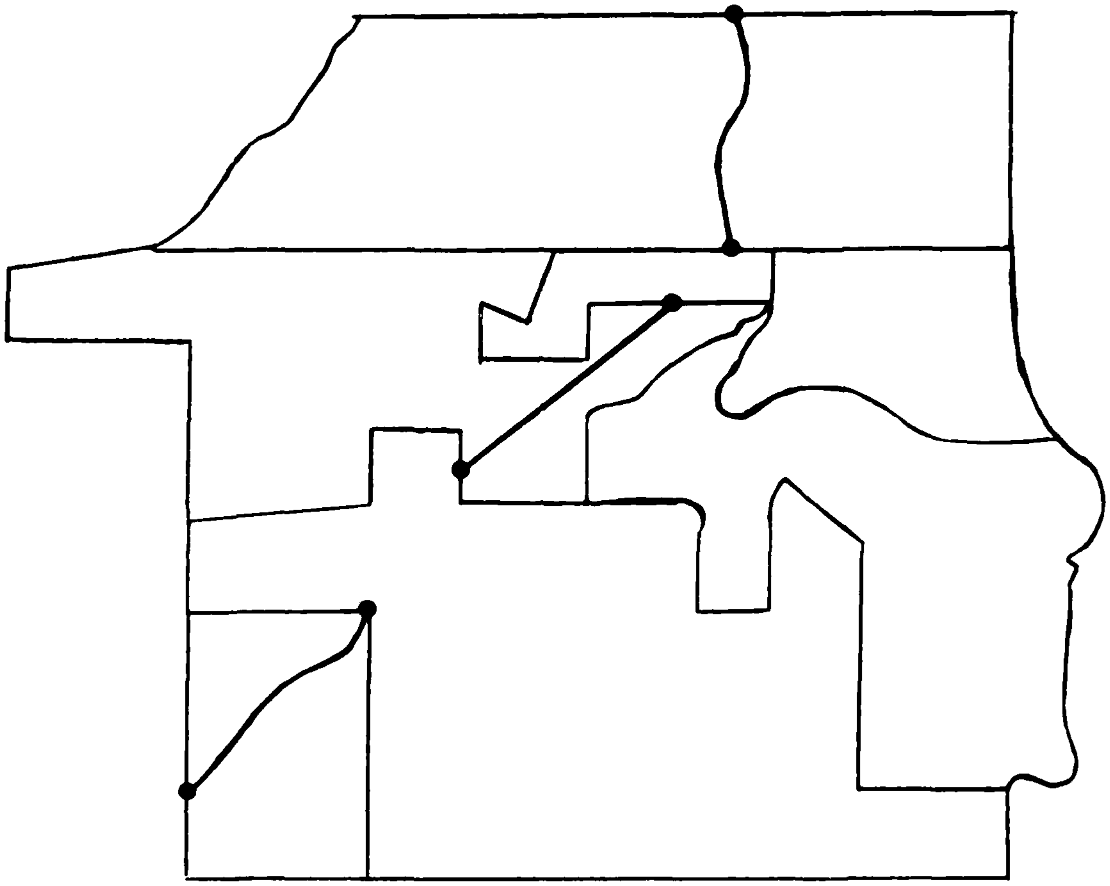


Figure 2.24. Lines clipped to the interiors of regions.

CHAPTER 3

IMAGE HANDLING

3.1 The Need

The system described in the previous chapter has the capability of storing, displaying, modifying, and analyzing cartographic information in the form of points, lines, regions, and related attributes. The source of this information is typically manual digitization coupled with keyboard entry. The data may also be read in from a digital file if one exists. In either circumstance, there is no guarantee that the information is correct or up-to-date.

One valuable source of information for validating and updating the locational aspects of cartographic data is aerial photography. Although some data types are not evident in aerial photography (e.g. administrative boundaries) many are. Typically rivers, roads, houses, and some aspects of land use and land cover can be detected. Therefore it would be useful to be able to overlay the existing vector data on a suitably registered photograph so that discrepancies could be observed directly and appropriate changes made. Adding this capability to the system was one of the tasks in the current project.

3.2 Imagery and Picture Files

Aerial photography of the Healdsburg California region was obtained from the EROS Data Center in Sioux Falls, S.D. It consists of eight black and white photographs taken 13 July 1979 between 12:24 and 13:10 local time. A photocopy of a portion of one of these images is shown in Figure 3.1.

This imagery was visually inspected to find the region that coincided with portions of the digital database. It was digitized using an Eikonix photodiode camera system. The Eikonix produces pixel files of up to 2048 x 2048 pixels. Each pixel is assigned an integer value between 0 and 255 which is a measure of the quantity of light captured by the photodiode. The right window sizes and pixel resolution for our purposes were determined by trial and error with the camera system.

The display device used on this contract was a Sanders/Calcomp Raster Monitor with a screen size of 720 by 1024 pixels and four bit planes of picture memory. It can display 16 colors or 16 distinct shades of gray. In order to display the pixel files generated by the camera system, the original pixel values (which ranged from 0 to 255) had to be compressed into the range from 0 to 14. This was done by a histogram smoothing technique which automatically mapped the original pixel values into the range from 0 to 14 in such a way that each of those values was assigned approximately the same number of pixels. The value 15 was reserved for the vectors that were eventually to be superimposed on the image. The smoothed picture file that resulted from this process was suitable for display on the graphics monitor. Each picture file was a direct access, unformatted file distinct from the AGIS data base.

3.3 Registration

A picture file was registered to the AGIS data base by a simple, two-point procedure. First the picture file and the map file are inspected separately to find common points in both. Road intersections, for instance, are usually apparent in both files. Two well separated points are selected and their (X, Y) positions in the AGIS file are determined through the AGIS query software. Then the picture file is displayed. The data tablet is used to identify the pixel locations in the image that correspond to the two AGIS (X, Y) positions. The resulting correspondence, which has the form (X_1, Y_1, row_1, column_1, X_2, Y_2, row_2, column_2), is stored in the picture file header record. When the two files are displayed together, affine transformations are used to make the two points from both files coincide exactly.

The procedure for digitizing does not guarantee that north will be directly vertical through the picture file. Yet the AGIS system displays vector data with north directly vertical unless the operator requests a rotation. Therefore the angular rotation necessary to overlay the two registration points can be accomplished in two ways: rotate the picture using pixel averaging methods so that north is up or rotate the map file to align it with the picture file. The system, as implemented, allows the operator to select either of these options.

The pros and cons are as follows. Rotating the picture by averaging pixel values can cause some blurring of the picture, particularly if the rotation angle is small (e.g. 1 or 2 degrees). Once the picture has been rotated so that north is vertical, no vector rotation is needed to overlay the map data on the image. This is advantageous since the AGIS edit functions only work on non-rotated files. Leaving the picture file intact and rotating the vectors results in a somewhat cleaner image for small angles of rotation but leaves a vector file that cannot be directly edited.

It may be the case that the operator would like to change the map data so that it corresponds to what is shown in the image. For example, areas of new construction or changes in natural features after a flood or earthquake could be entered directly from the image to the map file through AGIS edit functions. Therefore, rotating the picture would be the appropriate option in this case.

3.4 Display

AGIS allows the operator to call up the file in which an image is stored. Map data from the AGIS map files for that image area can then be overlaid. The operator is required to select which data types he wishes to see. Scaling and registration are handled automatically using the information stored in the header record of the image file. Figure 3.2 shows the display of a picture file before the vectors are superimposed. Figure 3.3 shows the same display with the vectors present. Figure 3.4 shows the vector file without the picture. Figure 3.5 shows the result of the following query highlighted and displayed on the picture (the result has been further highlighted on the photo for reproduction purposes).

Query: SHOW "REGIONS" <LANDUSE AREAS> REGIONS/WHERE[SID EQ "ACC"]
(Show all landuse areas that have a code of ACC)

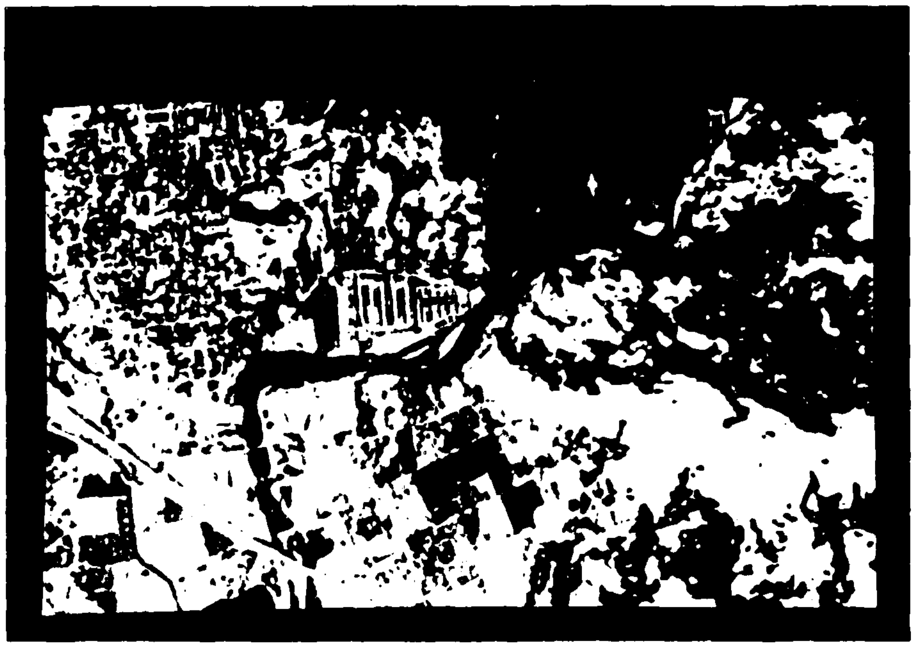


Figure 3.2. A Picture File Displayed by AGIS.

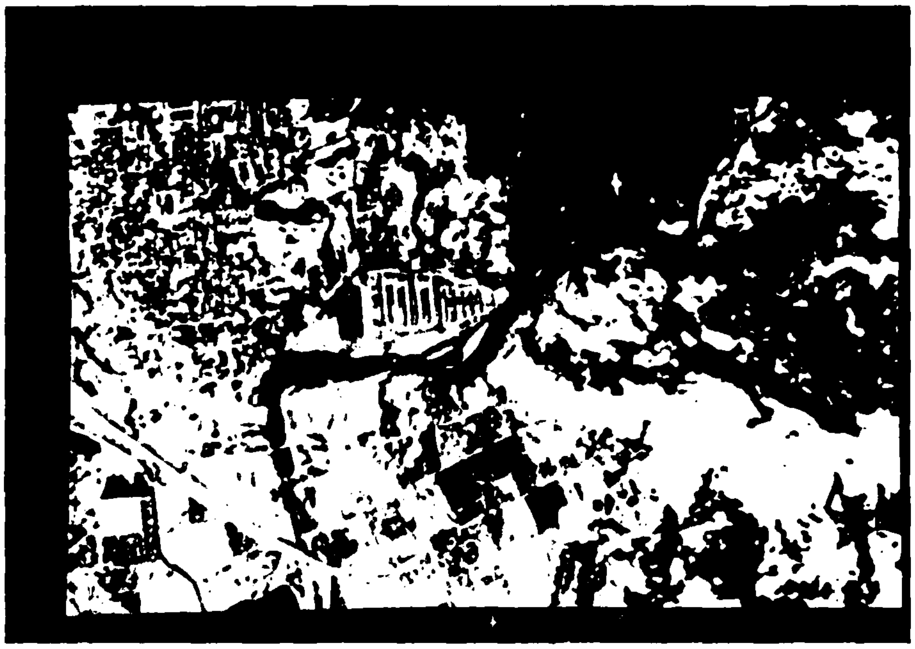
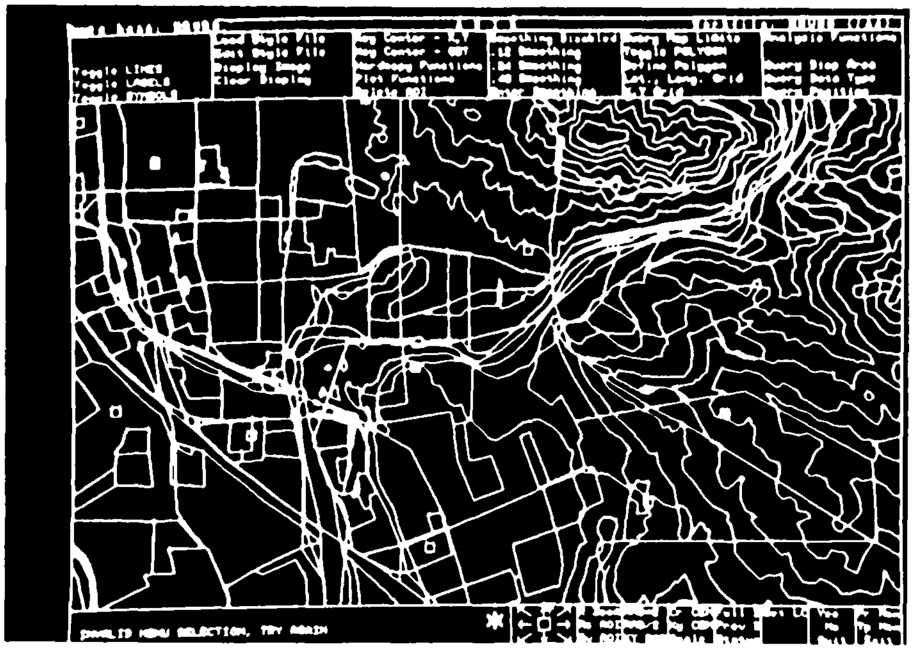


Figure 3.3. The Same Picture with Vectors Overlaid.



INVALID MENU SELECTION, TRY AGAIN

Figure 3.4. The Vector File from Figure 3.3 without the Picture.

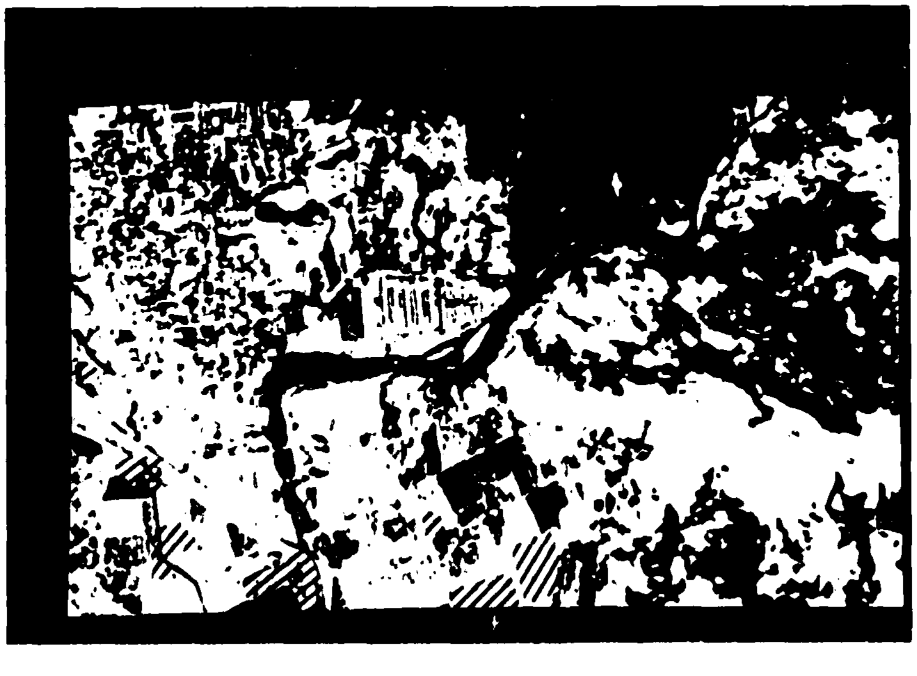


Figure 3.5. Query result overlaid on picture.

CHAPTER 4

WORKING WITH THE SYSTEM

4.1 Overview

When the programs required to perform the tasks for this study had been developed, a test database was built. It contains line, point, and region data for a single area, the Healdsburg Quadrangle. Imagery covering the test area was also included.

This chapter contains descriptions of how the database was built and of how the operator can perform various operations on the data using AGIS.

4.2 AGIS

The AGIS mapping system has full capabilities for data input, output, and analysis. The primary user interface is through menus which appear on the graphics screen. AGIS runs on a DEC VAX 11/780 under the VMS operating system. Digitization is done using a MEGATEK 7210 calligraphic refresh display and a SUMMA-GRAPHICS ID-48 digitizing table with a 13 button cursor. Plots are made on a Houston Instruments CPS-15/6 plotter. For display of imagery, a Calcomp Sanders color raster display is used.

4.3 Input

4.3.1 Map File Creation

The map file was created in a POLYCONIC projection (during Phase 1, data was input in the Lambert Conformal projection - this was changed so that the map data projection would match the digitized quadrangle maps). For every map sheet that was digitized, the operator was prompted to enter two registration points of 38° 35'N, 122° 52'30"W and 38° 37'30"N, 122° 50'W. By knowing the latitude and longitude of each registration point and by their corresponding digitizer X and Y coordinates, the proper scaling parameters were calculated to convert each digitized coordinate pair into a GBT address before placing the record into the map file. A GBT universe of eleven GBT digits was selected to provide a covering for the map area. Given the latitude and longitude bounds of the map area and the size of the GBT universe, a resolution of .6532 meters between adjacent hexagons provided greater accuracy than the digitizer input of 3.077 meters between points.

4.3.2 Manual Digitization

The map data was loaded by manually digitizing three mylar map overlays and a paper map covering the common area of the Healdsburg Quadrangle. The mylar maps described the three classes of region data: (1) landuse areas, (2) 100-year flood plain, and (3) the elevation contours. From the paper map, four classes of line data were digitized. Those classes were (1) roads, (2) rivers and streams, (3) power lines, and (4) railroads. Point data was also taken from the paper map. Within the single point class there were twelve types of point data: school, hospital, armory, gaging station, water tank, building, church, benchmark, substation, recreation center, well, and dam.

In digitizing, the operator first gave the system a class and layer specification. Every X, Y coordinate pair was converted into a GBT address and used to form a node in that class and layer. Line nodes entered in sequence were connected together by the software to form a line network with each node pointing to its neighbors. The data from the three mylar sheets was entered as one layer. The line data constituted a layer. All point data was placed in a third layer. When all the data had been manually digitized, the map file produced contained 8 classes in three layers.

The AGIS software integrated these classes into a single map file as the data was loaded. As each new record was added to the data base, it was automatically determined if data already existed at the same GBT address. In such a case the data records were combined to form a node with composite data characteristics. For example, a node at location A which is digitized as a landuse boundary might also be input as part of a road line. When the data was loaded there would be two nodes at location A, one from layer #1 class "landuse" and one from layer #2 class "road". These two records would be combined during integration into a single nonhomogenous node which carried both sets of properties. In contrast, a homogenous node is one that belongs to only one class and layer. The AGIS mapping software was designed and built to handle nodes that are either homogenous or nonhomogenous with regard to the class and layer.

The digitization process was quick and relatively few errors occurred. The operator had basic menu controlled editing functions to start, continue, or erase a line string. Individual nodes could be inserted, deleted, or moved. The system automatically connected line endings to the nearest node within a small radius. This feature was useful at line intersections. Figures 4.1 through 4.9 are plots of the individual classes and layers taken from the AGIS test database.

4.3.3 Verification of Region Topology

The region verification program was run on the three region classes: flood plain, landuse, and elevation. The program is relatively slow and was run as a batch process. At the end of this step, the database contained topologically correct region classes to which attribute data could be assigned.

4.3.4 Attribute Assignment

In order to assign attributes beyond layer and class to the spatial data contained in the map file, the operator first displays a selected layer and class. If the data to be assigned is for regions, the operator is prompted to pick a point inside the region. The software finds and traces the boundary as was described in Section 2.10. The operator is then asked to enter any additional information such as subject identification. In this way, each of the landuse areas was assigned a landuse code, each contour was associated with an elevation, and the 100-year flood plain was identified.

For line classes, the operator defines the line feature by selecting its start and end nodes. The software traces the line between them, asking the operator for directions at line junctions. The operator is prompted to enter subject identification. The software then constructs the appropriate signposts.

Point features are assigned by selecting them and entering the attribute information when prompted.

The attributes which were assigned for the test database are shown in Table 3.

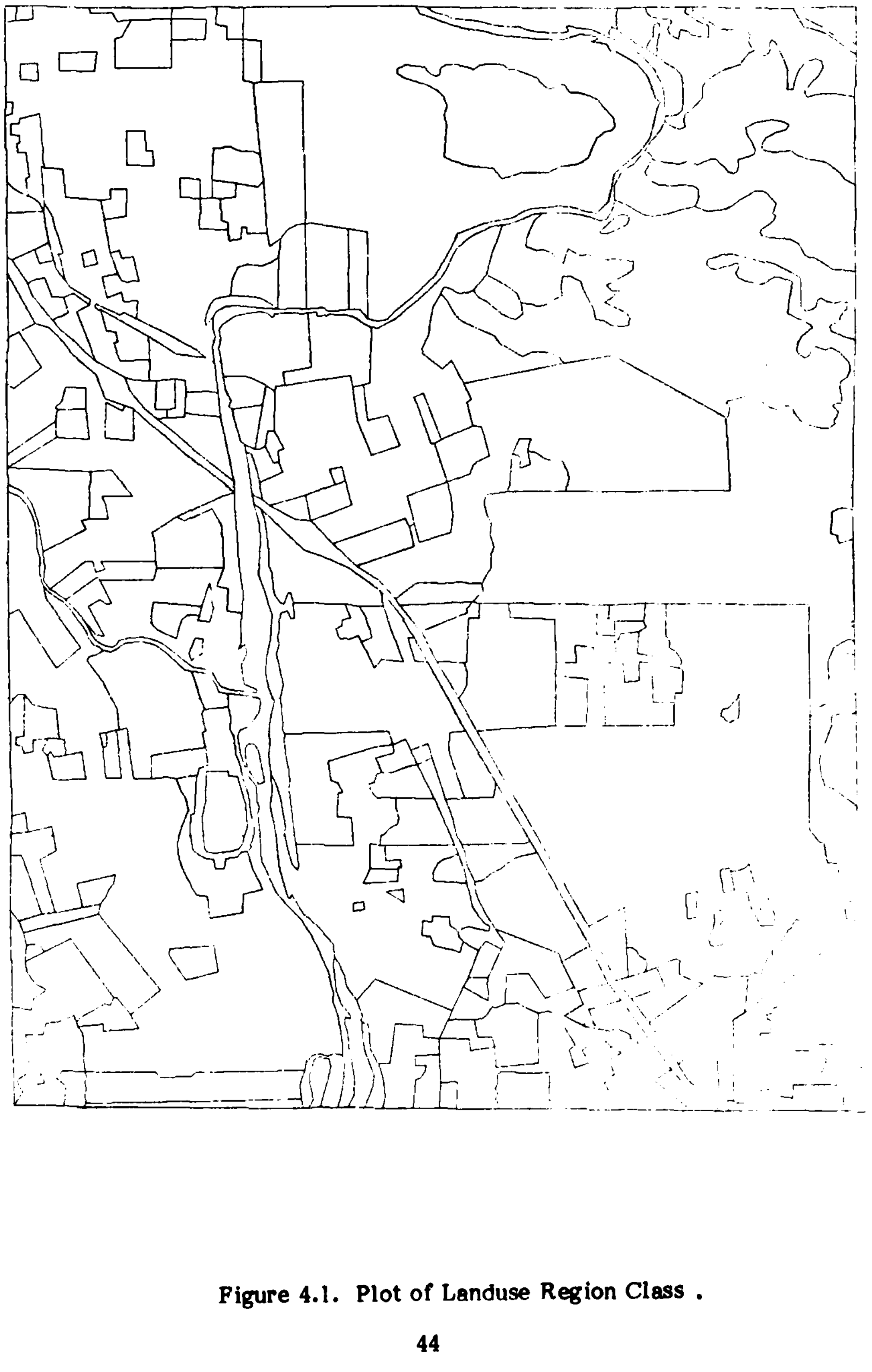


Figure 4.1. Plot of Landuse Region Class .

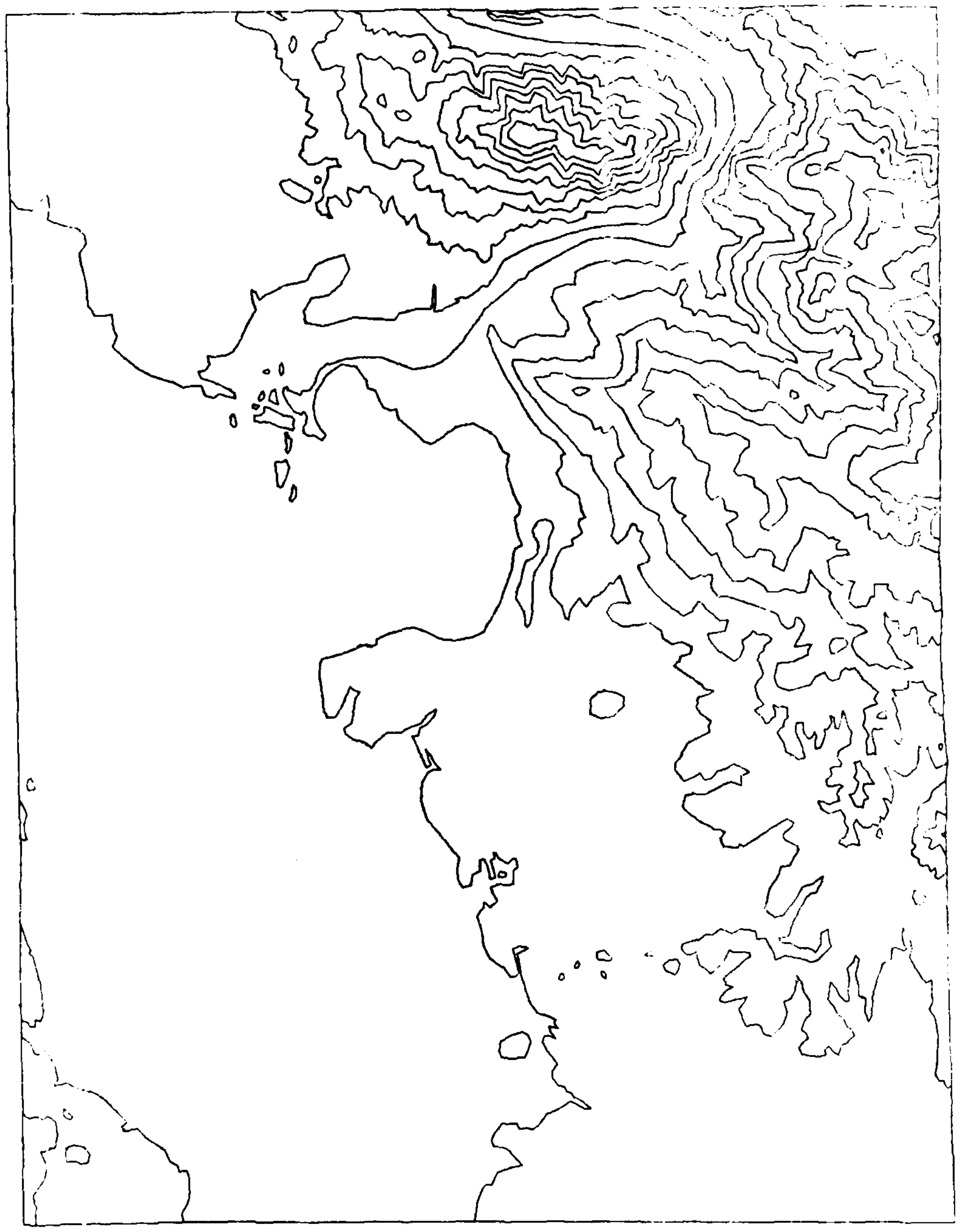


Figure 4.2. Plot of Contours Region Class

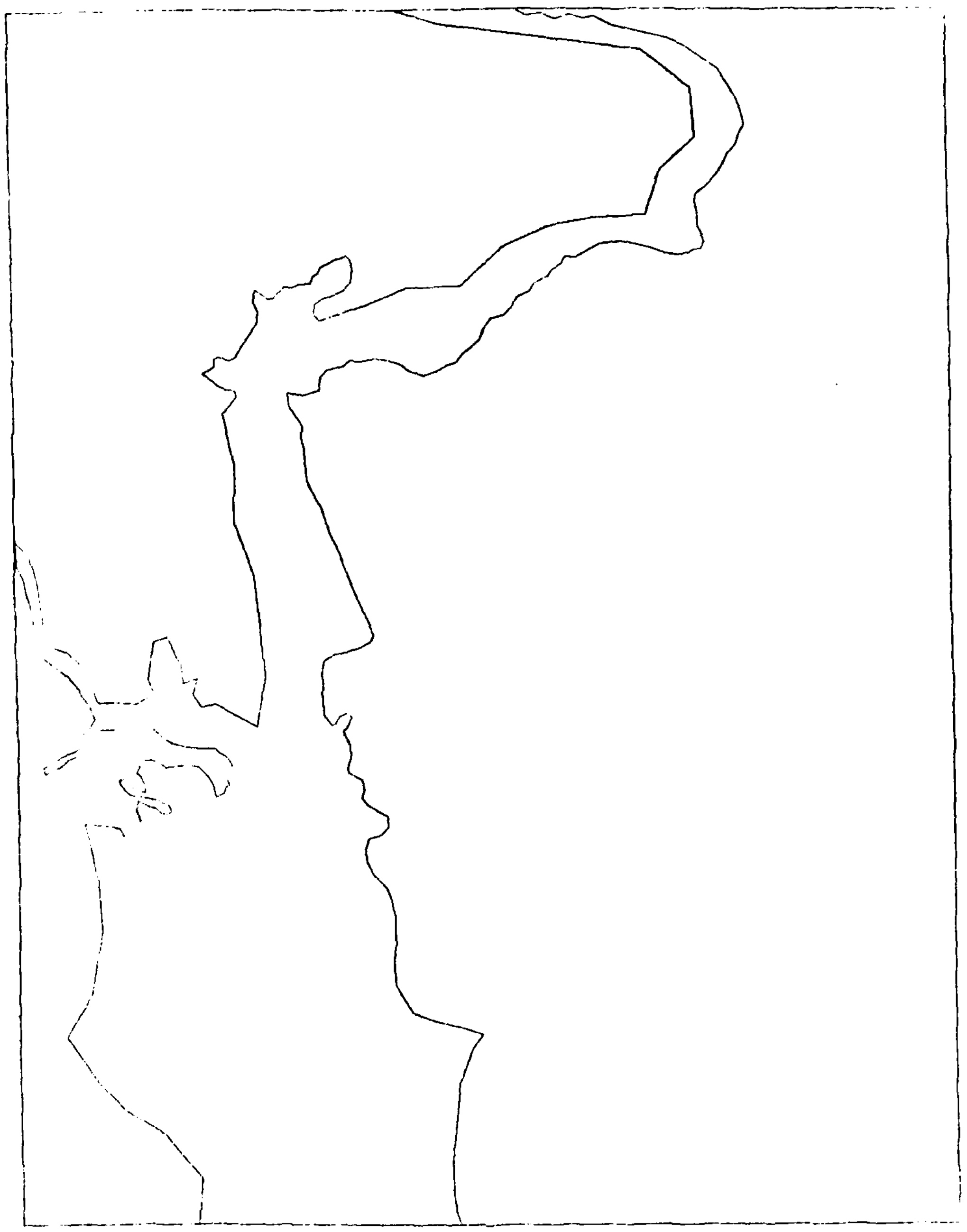


Figure 4.3. Plot of 100 Year Flood Plain Region Class

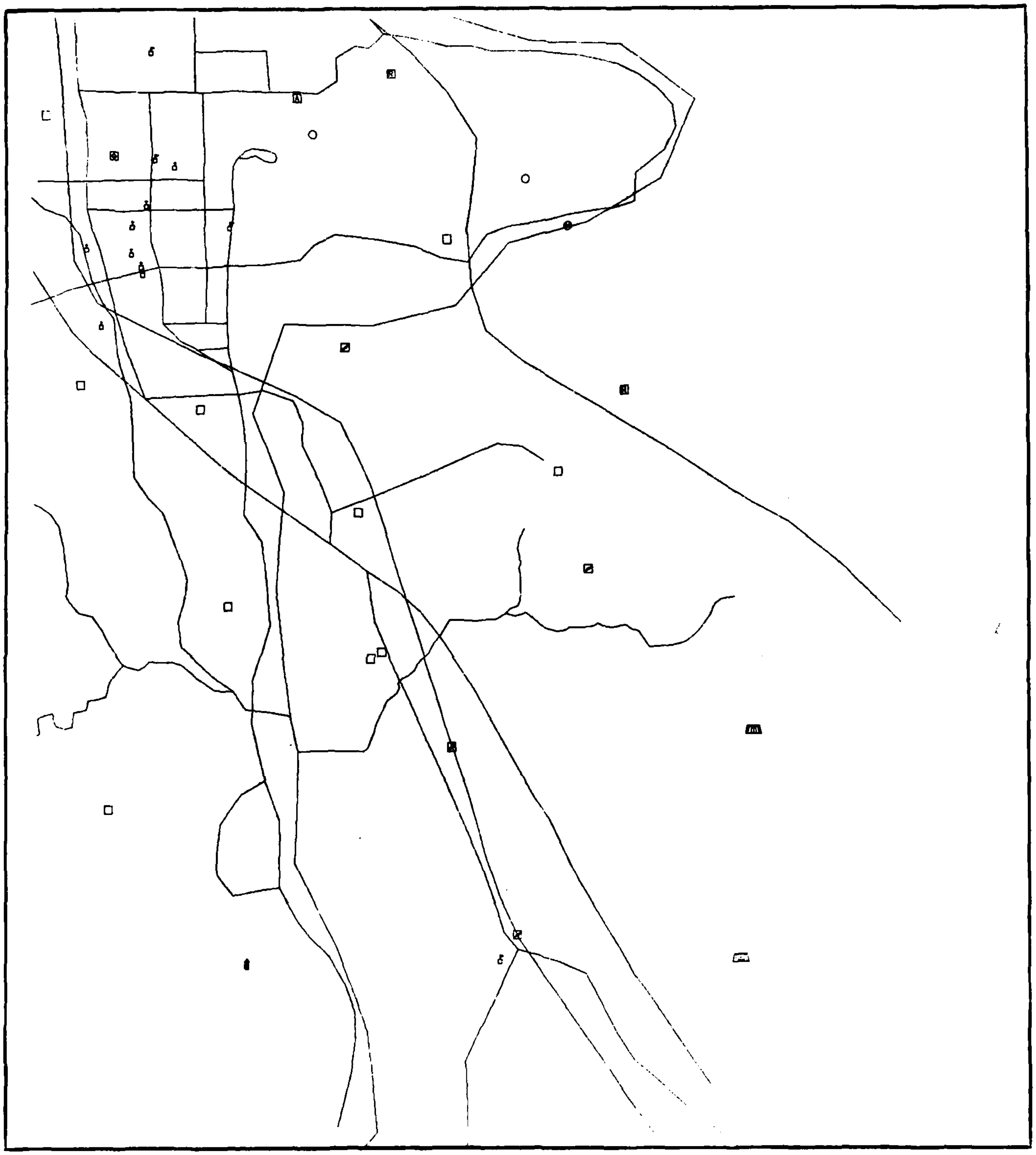


Figure 4.4. Plot of Combined Point and Line Layers

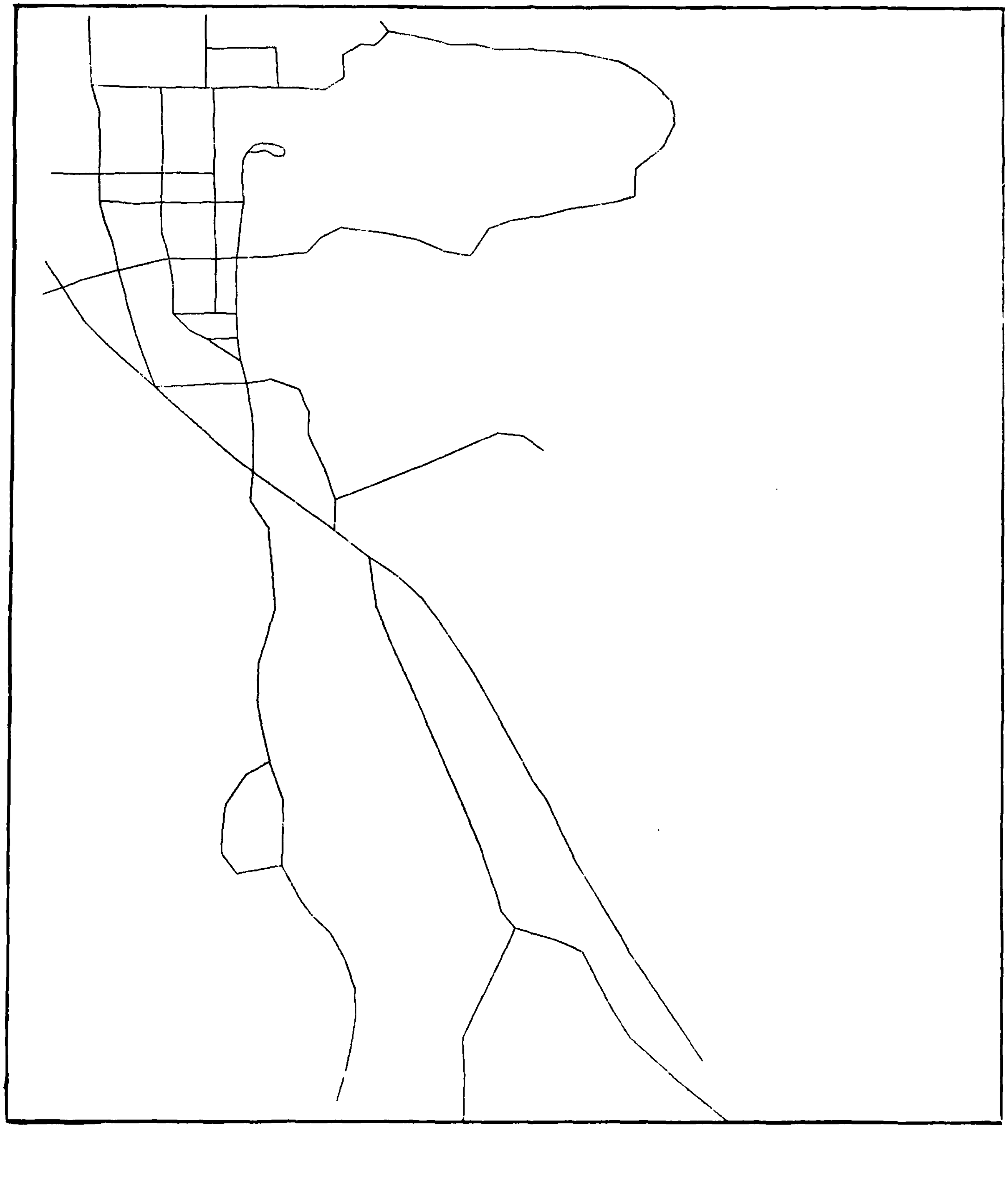


Figure 4.5. Plot of Roads Overlay

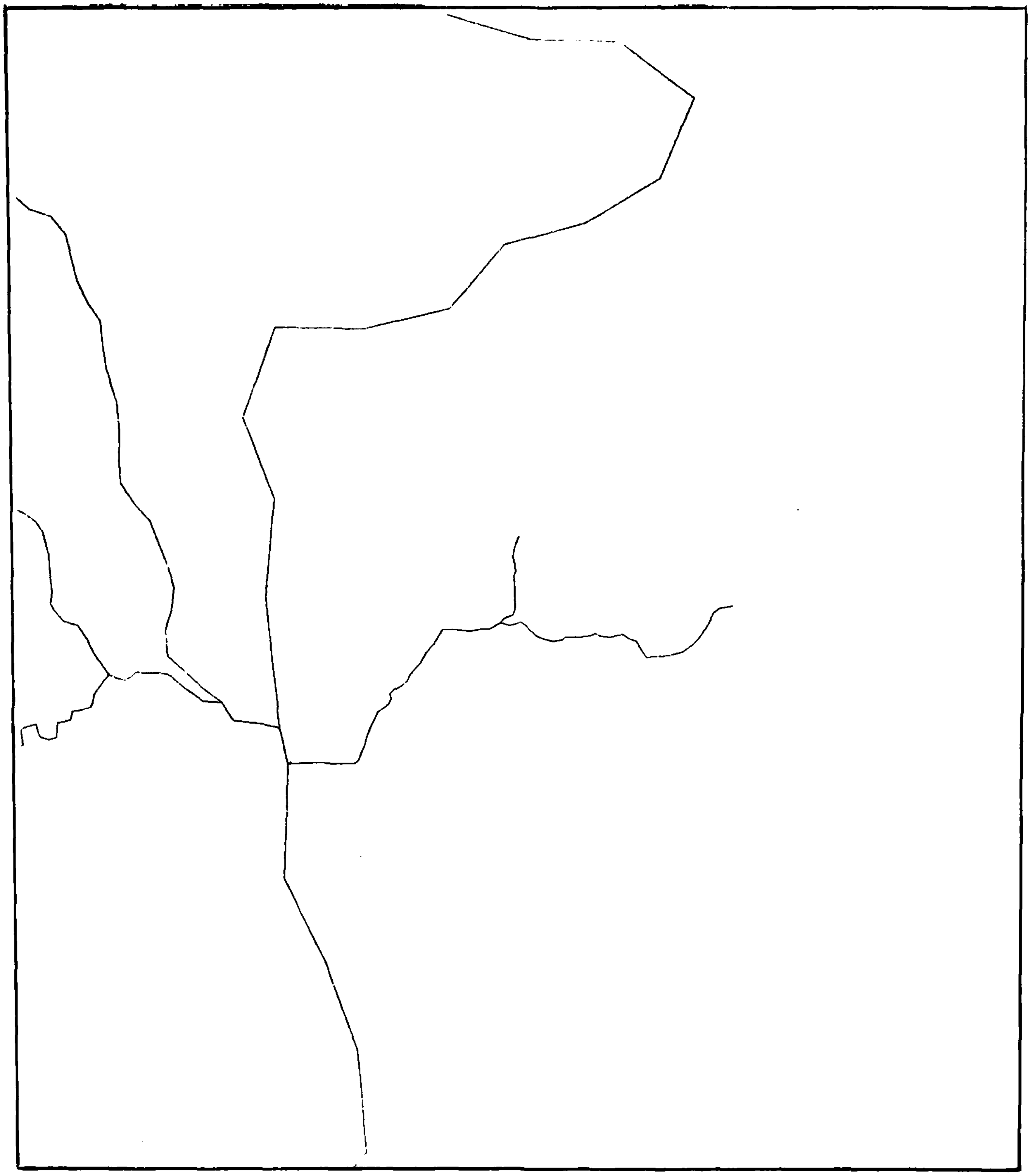


Figure 4.6. Plot of Rivers Class

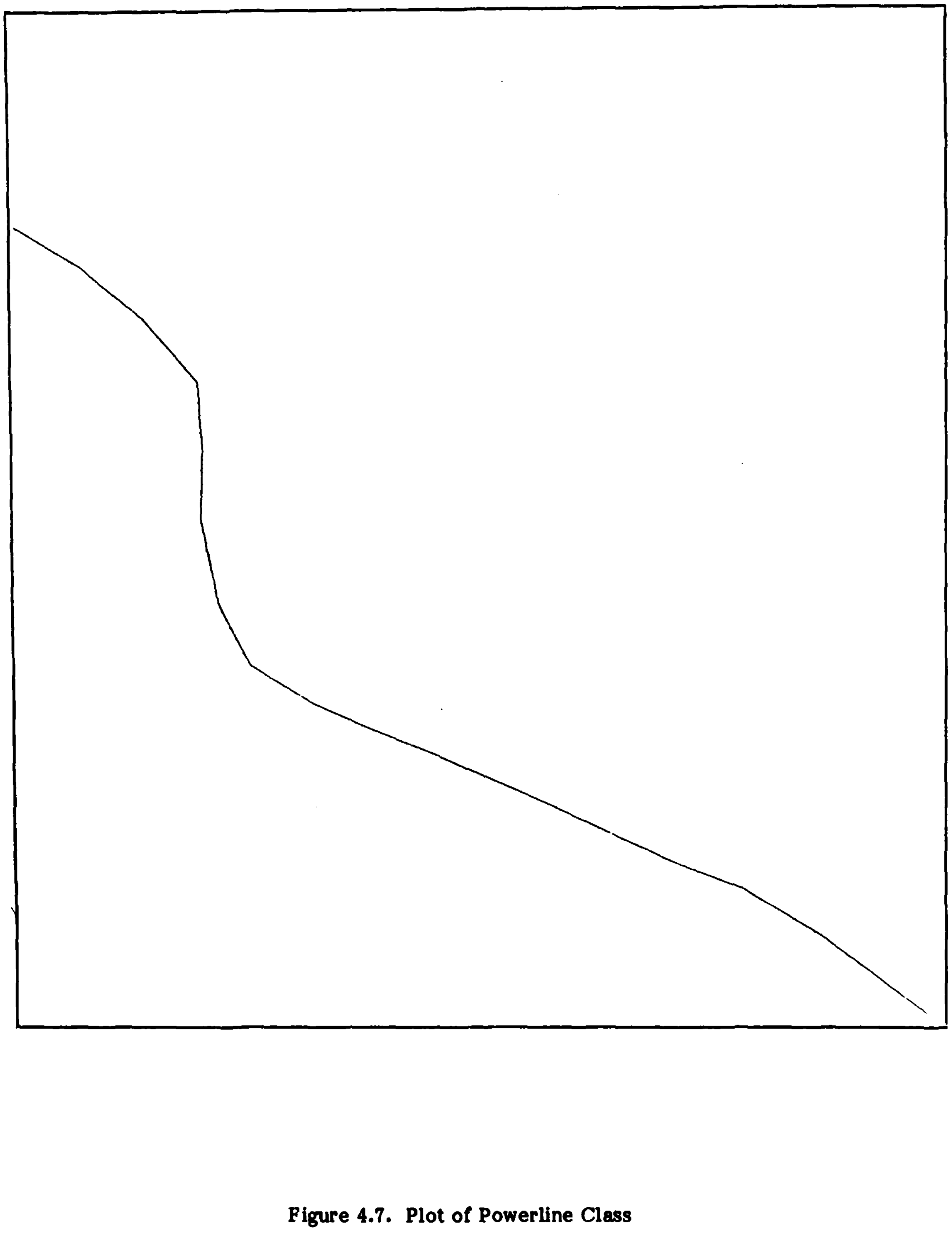


Figure 4.7. Plot of Powerline Class

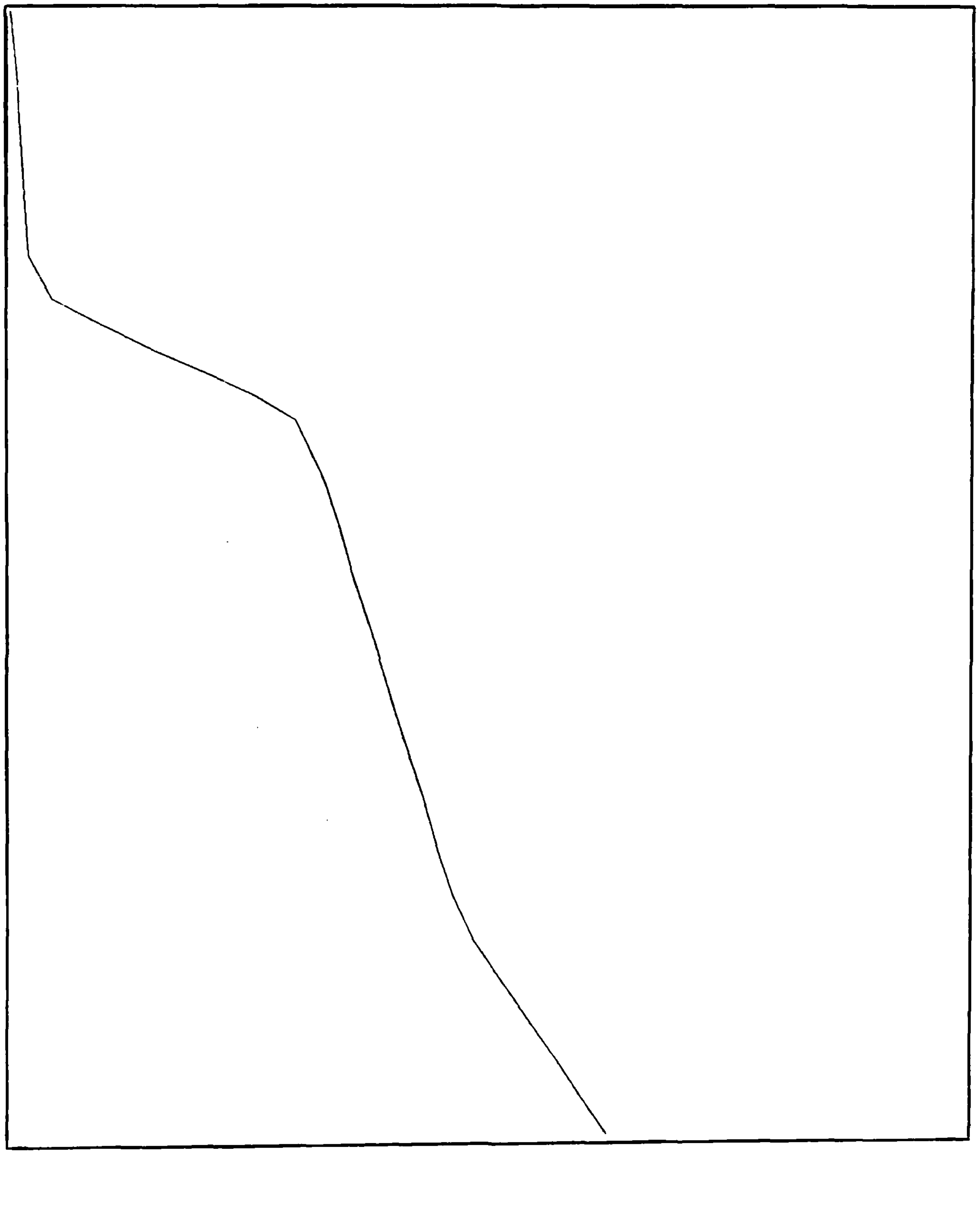


Figure 4.8. Plot of Railroads Class

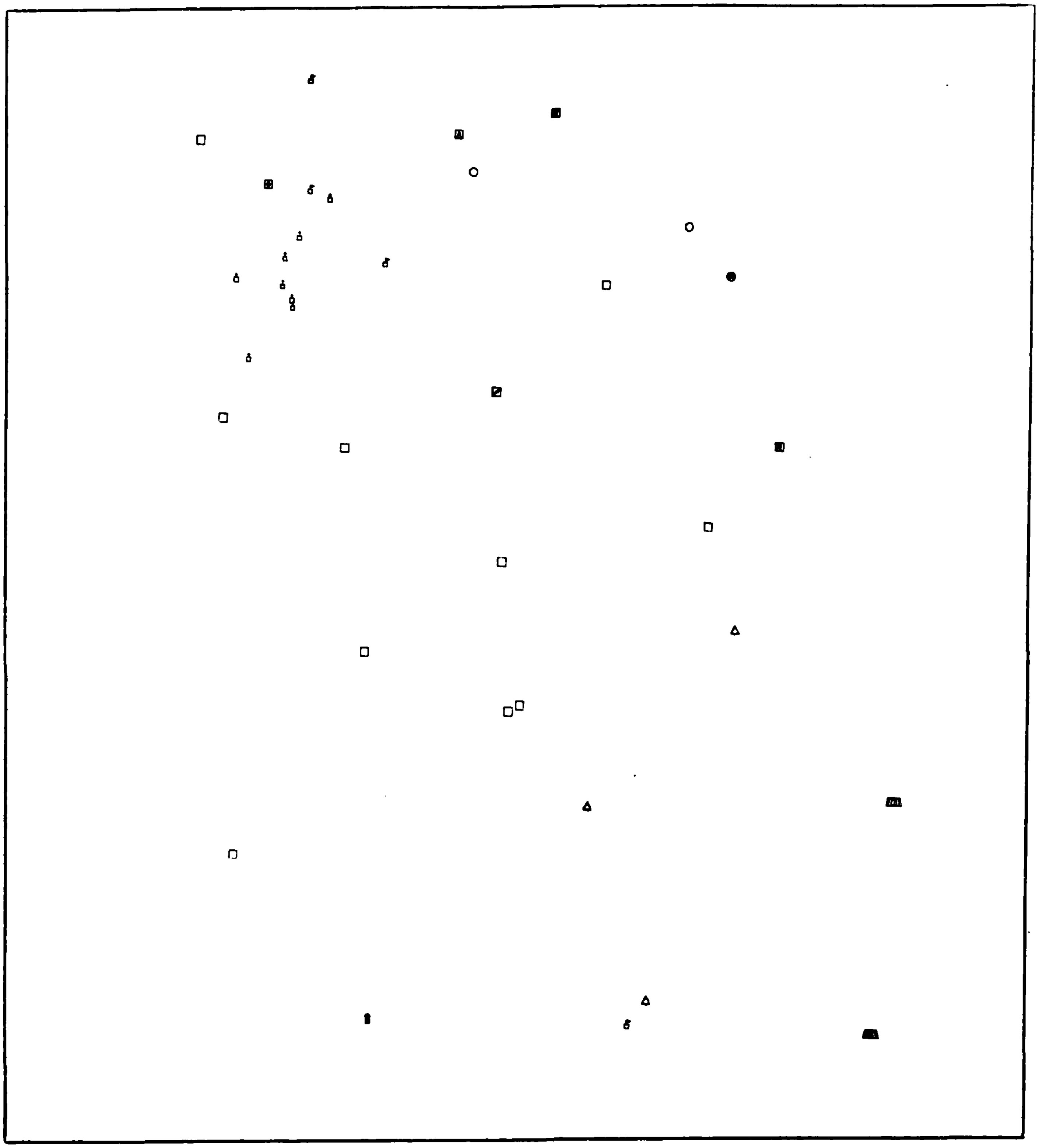


Figure 4.9. Plot of Point Data Layer

Data Description	Feature Type	Layer Name	Class Name	Subject ID Description
Landuse Overlay	Region	Regions	Landuse	Landuse Code
100 Year Flood Plain Overlay	Region	Regions	Flood Plain	Flood Plain/ Not Flood Plain
Elevation Contours Overlay	Region	Regions	Elevation	Elevation Value
Roads	Line	Lines	Road	Road 1 - Road 18
Rivers and Streams	Line	Lines	Rivers	Stream 1 - Stream 4
Powerlines	Line	Lines	Powerline	Powerline
Railroads	Line	Lines	Railroad	Railroad
Schools	Point	Points	All Points	School
Hospitals	Point	Points	All Points	Hospital
Armory	Point	Points	All Points	Armory
Gaging Station	Point	Points	All Points	Gaging Station
Water Tanks	Point	Points	All Points	Water Tank
Buildings	Point	Points	All Points	Building
Churches	Point	Points	All Points	Church
Benchmark	Point	Points	All Points	Benchmark
Substation	Point	Points	All Points	Substation
Recreation Center	Point	Points	All Points	Recreation Center
Wells	Point	Points	All Points	Well
Dam	Point	Points	All Points	Dam

Table 3. Organization of Test Data by Feature, Class, and Layer.

4.3.5 Database Statistics

Tables 4 and 5 show file sizes and process times for the test data base.

STORAGE REQUIREMENTS	# ENTITIES	# UNPACKED NODES
LANDUSE	184	2251
ELEVATION	72	4928
FLOODPLAIN	3	348
LINES	24	540
POINTS	36	36

FILE SIZES FOR THE DB - UNPACKED	
SPATIAL DATABASE	744/BLOCKS *
ATTRIBUTE DATABASE	28/BLOCKS

* 1 Block = 512 Bytes

Table 4. File Size.

DATABASE CREATION AND DIGITIZING

	CPU TIME	OPERATOR TIME
- LANDUSE	17.9 MIN	3.5 HR
- CONTOURS	34.8 MIN	3.3 HR
- FLOODPLAIN	2.5 MIN	.5 HR
- LINES	5.0 MIN	1.0 HR
- POINTS	2.0 MIN	1.0 HR
NODE EDITING	20.0 MIN	2.5 HR

ASSOCIATING ATTRIBUTES

	CPU TIME	OPERATOR TIME
- REGIONS	80.0 MIN	5.0 HR
- LINES	30.0 MIN	1.0 HR
- POINTS	30.0 MIN	1.0 HR

VERIFICATION

	CPU TIME	OPERATOR TIME
- LANDUSE	5.0 HRS	5.0 MIN
- CONTOURS	4.0 HRS	5.0 MIN
- FLOODPLAIN	1.0 HR	5.0 MIN

Table 5. File Creation.

4.4 Query Functions

The AGIS system has a variety of query and display capabilities. The query language was described in Chapter 2. A set of queries was outlined as part of this study. They were chosen as representative and as an appropriate demonstration set. They are listed in Table 6 (along with the English description). The results of these queries were stored in map files and then displayed . Figures 4.10 through 4.16 are plots of the results of the queries in Table 6. (Queries 8 and 9 are the two queries that could not be done during the Phase I study. They were attepted again with the same results.) Table 7 shows the accumulated elapsed and CPU time for each query.

```
1)  SHOW "POINTS"<ALL POINTS>POINTS/WHERE[SID EQS "BUILDING"]
    (Show all buildings)

2)  SHOW "POINTS"<ALL POINTS>POINTS/WHERE[SID EQS "CHURCH"].INTERIOR.+
          "REGIONS"<LANDUSE AREAS>REGIONS/WHERE[SID EQS "UCC"]
    (Show all churches inside landuse area UCC)

3)  SHOW "POINTS"<ALL POINTS>POINTS/WHERE[SID EQS "CHURCH"]+
          .EXTERIOR."REGIONS"<LANDUSE AREAS>+
          REGIONS/WHERE[SID EQS "UCR"]
    (Show all churches not in landuse area UCR)

4)  SHOW "LINES"<ROADS>LINES
    (Show all roads)

5)  SHOW "LINES"<RAILROADS>LINES.INTERIOR."REGIONS"+
          <LANDUSE AREAS>REGIONS/WHERE [SID EQS "UCR"]
    (Show all railroads inside landuse area UCR)

6)  SHOW "LINES"<RAILROADS>LINES.EXTERIOR."REGIONS"+
          <LANDUSE AREAS>REGIONS/WHERE [SID EQS "UCR"]
    (Show all railroads not in landuse area UCR)

7)  SHOW "LINES<ROADS>LINES.PLUS."POINTS"<ALL POINTS>POINTS/WHERE+
          [SID EQS "SCHOOL" OR SID EQS "HOSPITAL"]
    (Show all roads, schools, and hospitals)

8)  SHOW "REGIONS<FLOOD PLAIN>REGIONS/WHERE+
          [SID EQ "FLOOD PLAIN"].OR."REGIONS"+
          <LANDUSE AREAS>REGIONS/WHERE[SID NE "AVV"].AND.+
          "REGIONS"<CONTOURS>REGIONS/WHERE[SID EQ "100"]
    (Show all floodplain and landuse area that is not AVV within the 100' contour area)

9)  SHOW "REGION"<CONTOURS>REGIONS.AND."REGIONS"+
          <FLOOD PLAIN>REGIONS/WHERE[SID EQ "NOT FLOOD PLAIN"]
    (Show all contour areas outside the floodplain)
```

Table 6. Test Queries for AGIS system.

Query	Elapsed Time *	CPU Time *
1	0:05.85	0:00.86
2	1:17.09	0:10.87
3	1:50.12	0:17.18
4	1:45.13	0:05.28
5	8:29.04	0:12.22
6	6:11.39	0:16.71
7	0:29.08	0:04.28
8	N/A	N/A
9	N/A	N/A

* mm:ss.ss

Table 7. Accumulated Query Processing Time.

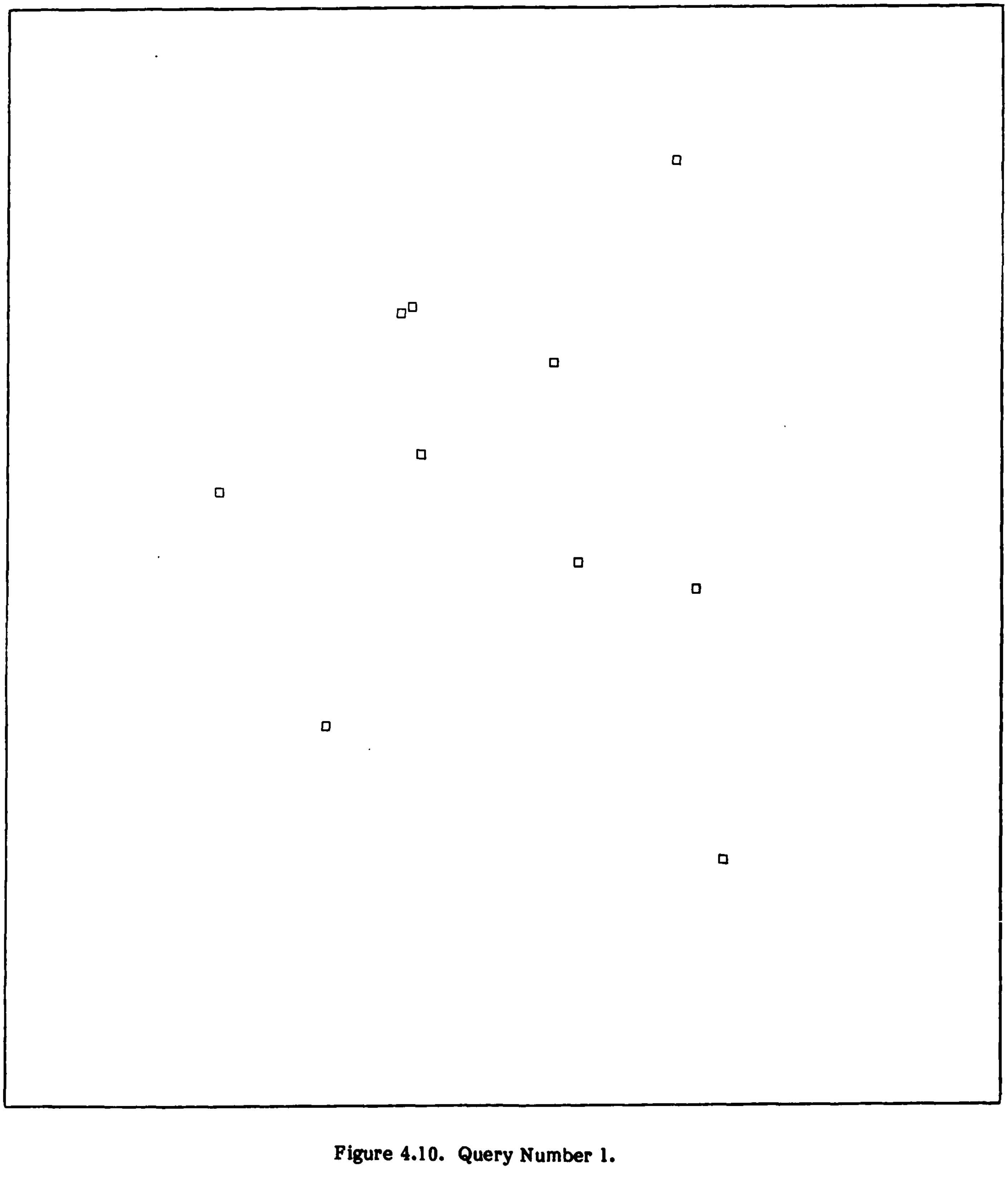


Figure 4.10. Query Number 1.

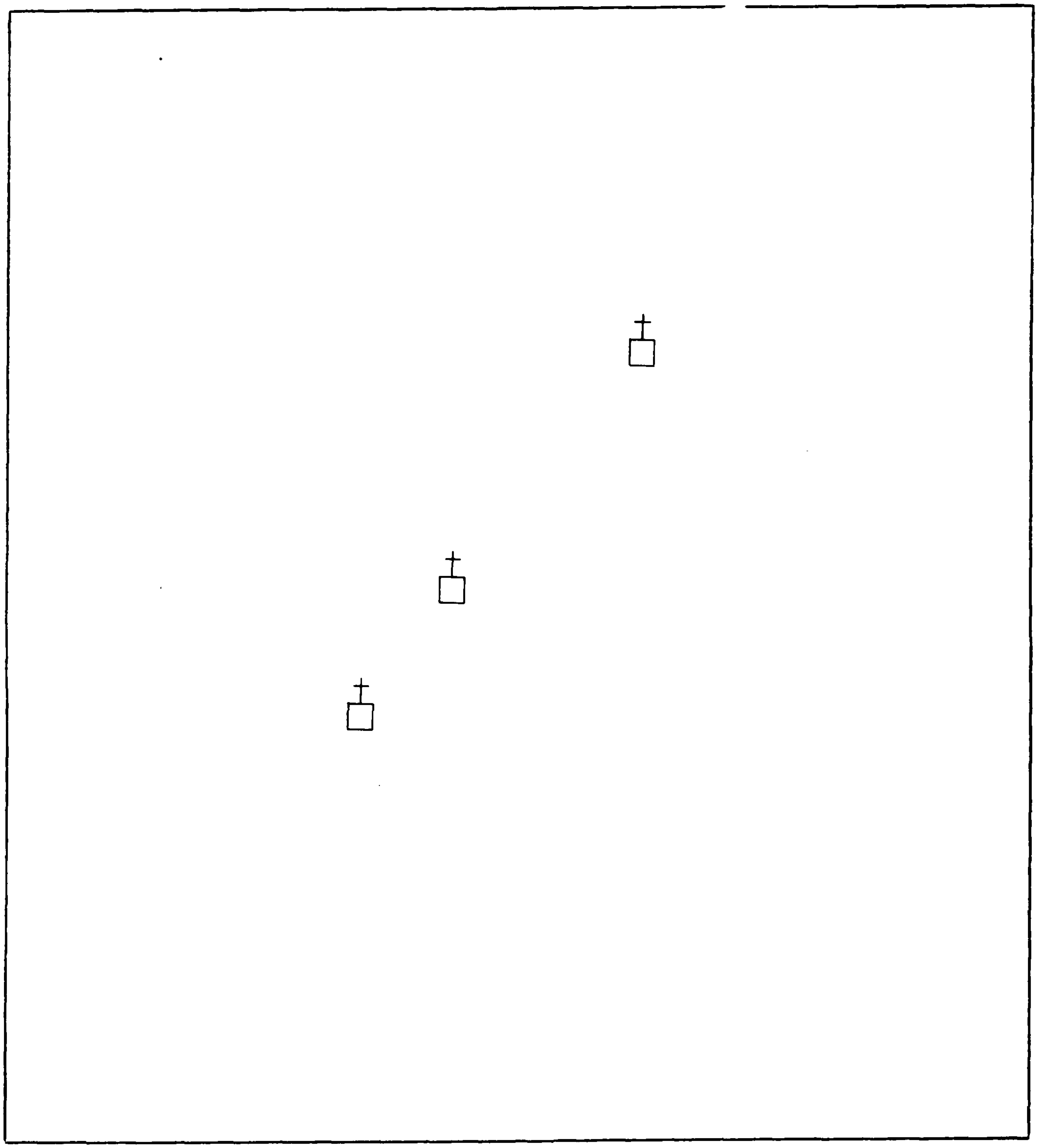


Figure 4.11. Query Number 2.

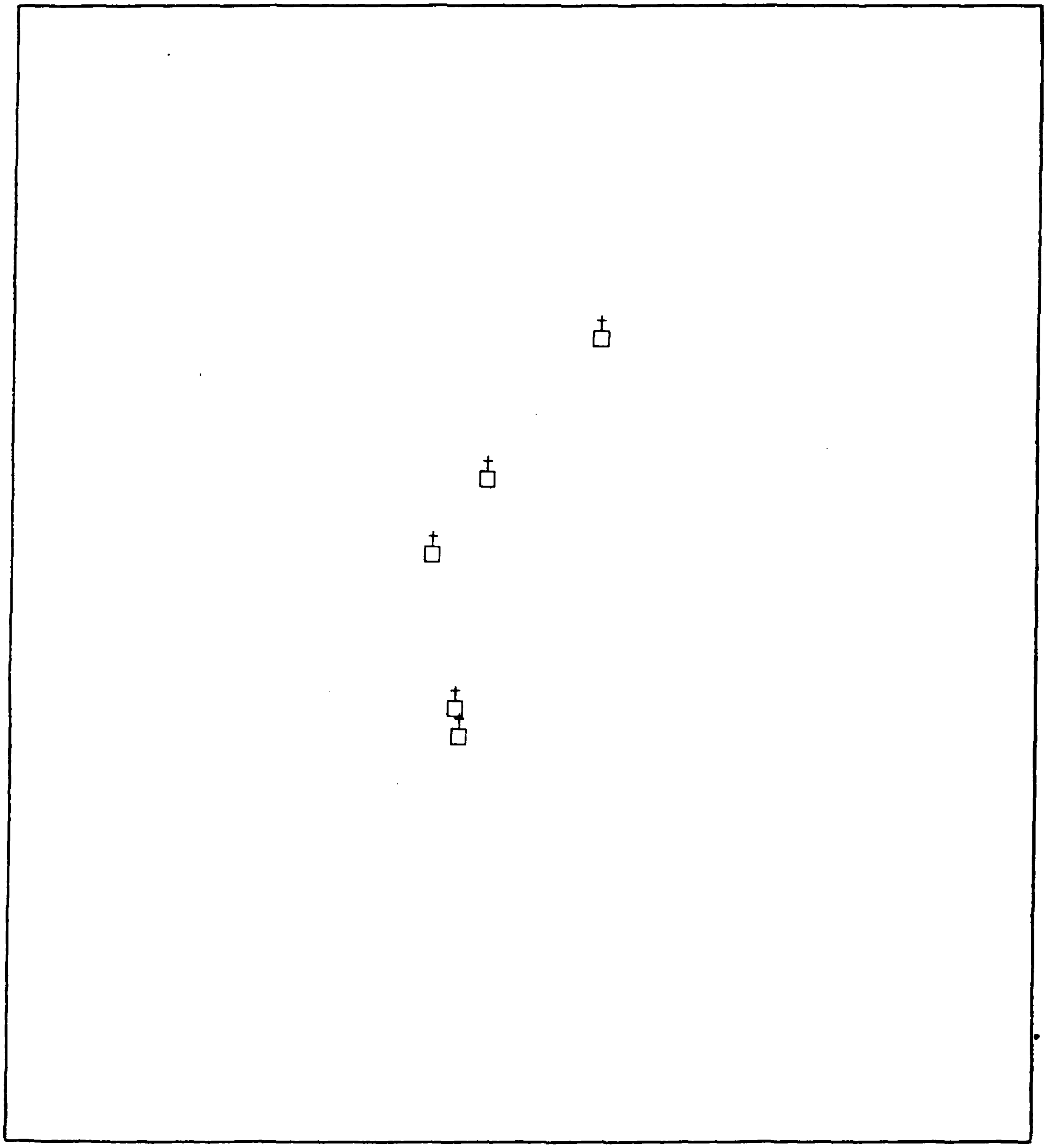


Figure 4.12. Query Number 3.

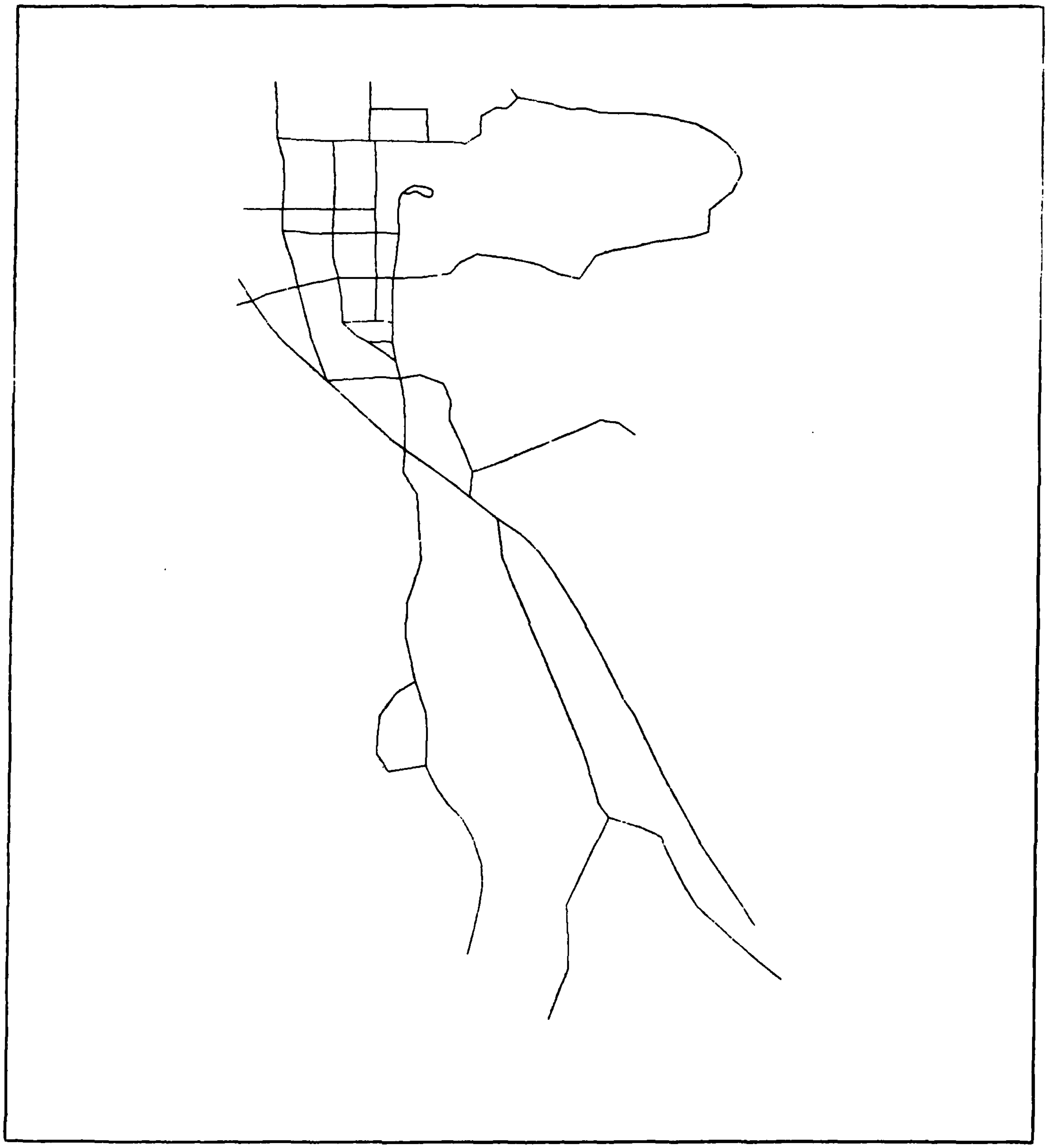


Figure 4.13. Query Number 4.

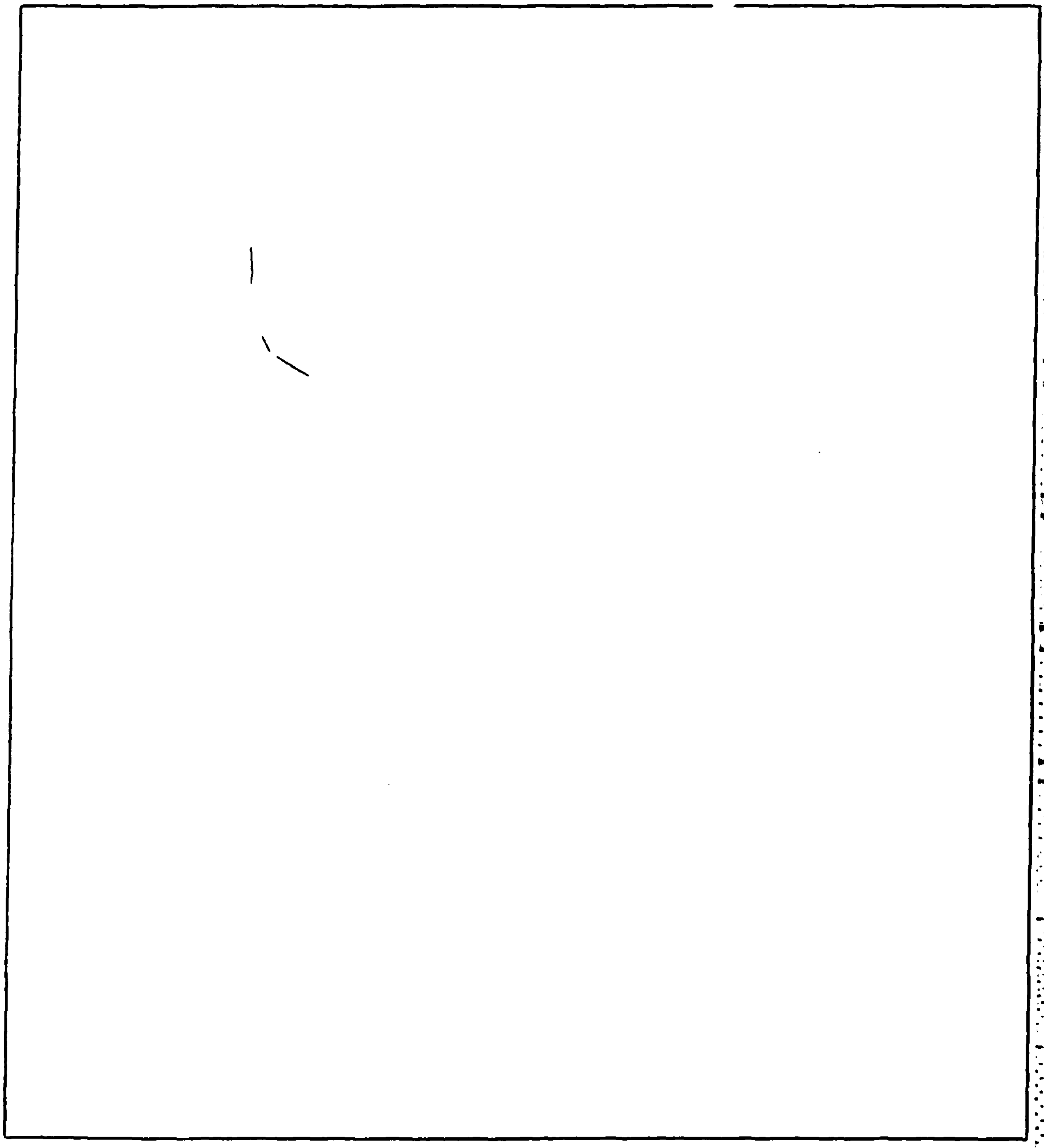


Figure 4.14. Query Number 5.

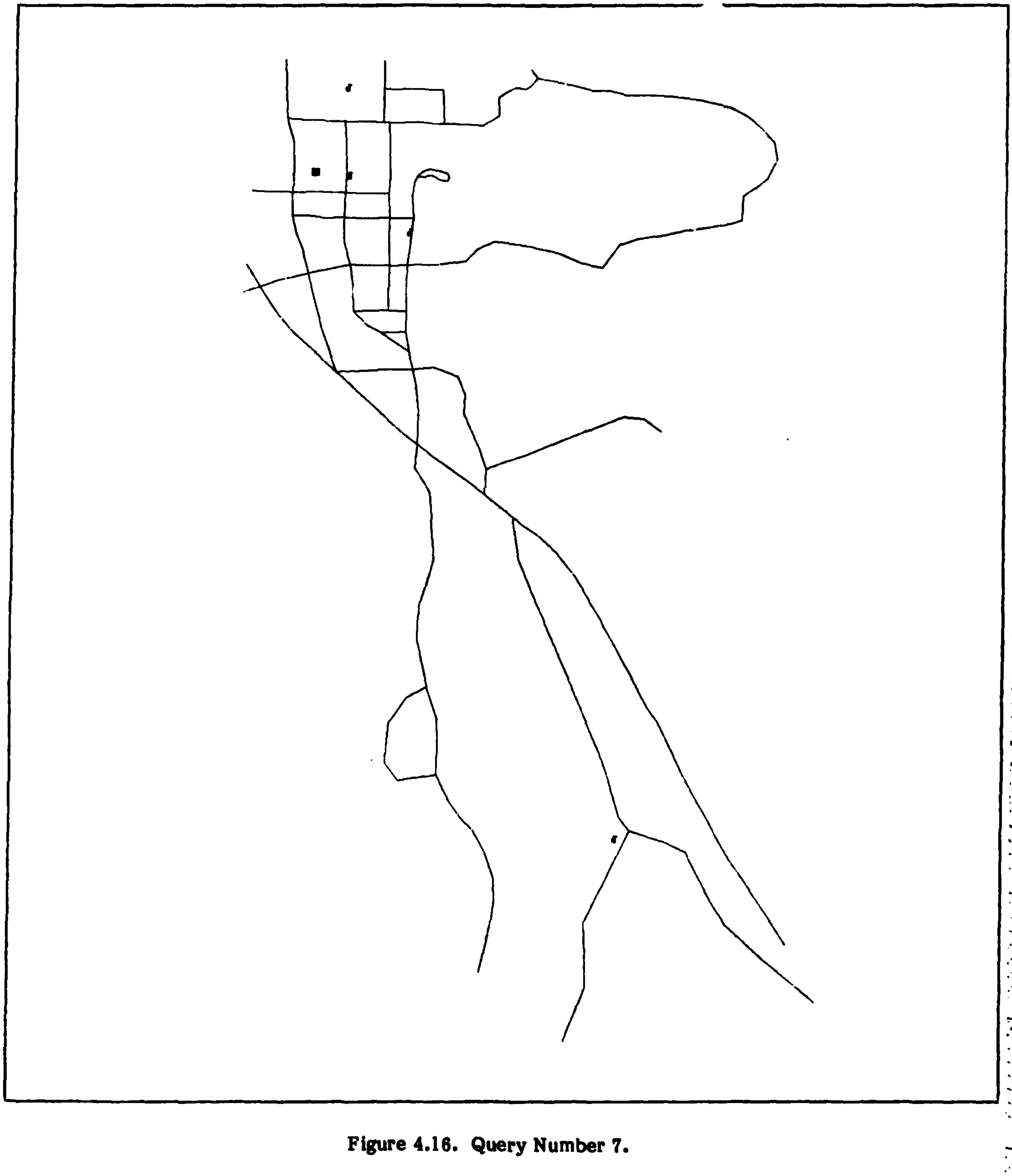


Figure 4.16. Query Number 7.

CHAPTER 5

RESULTS AND CONCLUSIONS

5.1 Overview

We were fortunate during the second year of this study in being able to work with a system which had been enhanced to overcome design difficulties uncovered during the first year. These enhancements were not part of this contract but had been made to support customer requirements for the AGIS product. In particular, the approach described in Chapter 2 for representing region data is much more unified and graph theoretic than that of the previous implementation (see Hexagonal Data Base Study, September 1983). The automatic association of islands was a fairly straight forward process with this new framework. In addition, the fact that the region handling algorithms had been strengthened facilitated the work involved in integrating line and point data.

The tasks for this study were of three types. The first was the integration of data types (point, line, and region) into one database. The second was to build a function to automatically identify and tag islands in region layers. The third set of tasks involved combining information from the map database with digitized aerial photographic imagery. The results of each of these tasks is discussed below.

5.2 Integrate Point, Line, and Region Data

AGIS, by making use of the GRAM data manager, has a natural means for combining data. The most difficult aspect of this task was the representation of linear features. The current implementation is adequate if care is taken in editing. Since middle portions of a line feature do not contain information about the existence or attributes of the feature, careless editing can result in linear features without the necessary signposts and which are therefore "lost". Additional software could be written to protect the files from this type of editing. The edit functions should be enhanced to automatically maintain signposts in all cases.

The system performed well on the queries which combined different data types.

5.3 Automatic Association of Islands

In the previous implementation, key information about the existence of islands in region layers was stored as attribute data in RIM. The current representation contains such information in the map file. This eliminated many problems in working with islands. The use of isthmuses seems to be a good idea. Having an automatic function to correct errors in topology and to insert isthmuses is also good. This process is relatively slow because of the algorithm which checks for unmarked line crossings. Comparing all possible pairs of line segments in a large file is time consuming. The algorithm does not take advantage of the GBT aggregate structure and it is felt that a much more efficient algorithm would result if the aggregates were used. Since this particular algorithm is used in several processes (for example, in queries), enhancing it would improve the entire system.

5.4 Imagery

The ability to examine data from the map database in conjunction with digitized imagery is useful in many applications. By overlaying the map data on an image, the data can be checked and updated directly in the computer. During this study, this capability was added to AGIS. Clearly, a key aspect of this function is the accurate registration of the image. The technique used was very simple: select two registration points and apply an affine transformation to make the image line up with the map data. For display purposes the image is rotated to show north vertical. This technique was adequate for the purposes of the study. It was found that the map data did not correspond with the aerial photography, in ways and to a degree which suggested the differences were not due to registration alone. Still the need for better registration is clear. Properties of the camera system, the photography (projection, distortion, etc.), and the map data should be taken into account.

REFERENCES

1. D. Lucas and L. Gibson, Automated Analysis of Imagery, Final Report on Contract Number F49620-82-C-0070, November 1982.

2. D. Lucas, "A Multiplication in N-Space," Proceedings of the American Mathematical Society, Volume 74, Number 1, April 1979

3. D. Lucas and L. Gibson, A System for Hierarchical Addressing in Euclidean Space, Martin Marietta Corporation, January 1980.

4. L. Gibson and D. Lucas, "Spatial Data Processing Using Generalized Balanced Ternary," Proceedings of the IEEE Computer Society Conference on Pattern Recognition and Image Processing, June 1982.

5. D.E. Knuth, "The Art of Computer Programming," Vol. 2, Addison - Wesley, Reading Mass., 1969.